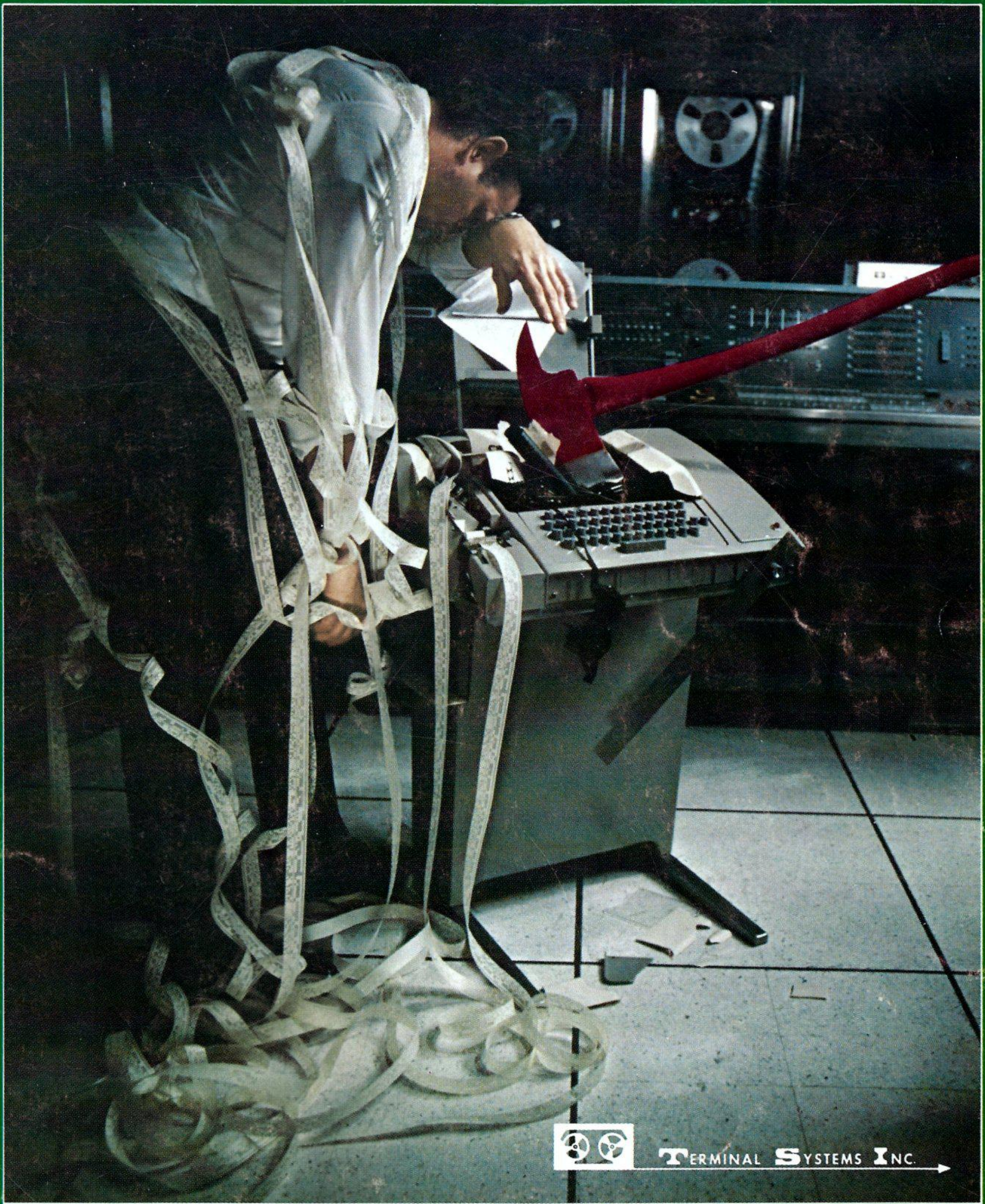
Price $1.50

SCCS INTERFACE

VOL. 1 ISSUE 4 MARCH 1976

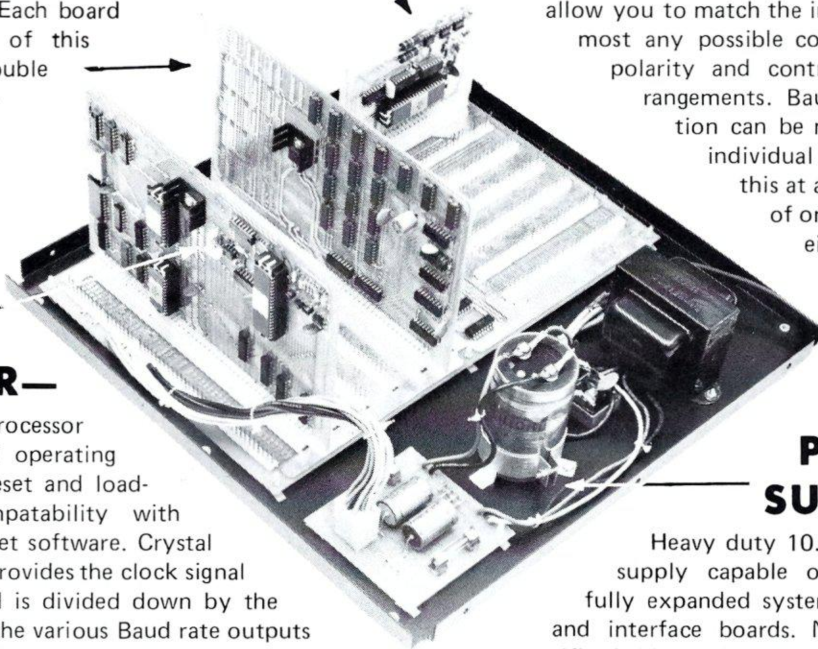
THE NATIONAL PUBLICATION OF THE SOUTHERN CALIFORNIA COMPUTER SOCIETY


TERMINAL SYSTEMS INC.

FEATURES

- Introduction to Microprocessor Technology
- Calculatin' Engines
- The Raw and the Cooked
- Serial Communications

WHY SETTLE FOR LESS— THAN A COMPLETE 6800 SYSTEM



MEMORY—

All static memory with selected 2102 IC's allows processor to run at its maximum speed at all times. No refresh system is needed and no time is lost in memory refresh cycles. Each board holds 4,096 words of this proven reliable and trouble free memory. Cost—only $125.00 for each full 4K memory.

INTERFACE—

Serial control interface connects to any RS-232, or 20 Ma. TTY control terminal. Connectors provided for expansion of up to eight interfaces. Unique programmable interface circuits allow you to match the interface to almost any possible combination of polarity and control signal arrangements. Baud rate selection can be made on each individual interface. All this at a sensible cost of only $35.00 for either serial, or parallel type

PROCESSOR—

"Motorola" M6800 processor with Mikbug® ROM operating system. Automatic reset and loading, plus full compatability with Motorola evaluation set software. Crystal controlled oscillator provides the clock signal for the processor and is divided down by the MC14411 to provide the various Baud rate outputs for the interface circuits. Full buffering on all data and address busses insures "glitch" free operation with full expansion of memory and interfaces.

POWER SUPPLY—

Heavy duty 10.0 Amp power supply capable of powering a fully expanded system of memory and interface boards. Note 25 Amp rectifier bridge and 91,000 mfd computer grade filter capacitor.

DOCUMENTATION—

Probably the most extensive and complete set of data available for any microprocessor system is supplied with our 6800 computer. This includes the Motorola programming manual, our own very complete assembly instructions, plus a notebook full of information that we have compiled on the system hardware and programming. This includes diagnostic programs, sample programs and even a Tic Tac Toe listing.

Mikbug® is a registered trademark of Motorola Inc.

SwTPC 6800
Computer System

with serial interface and 2,048 words of memory. $395.00

SwTPC 6800 COMPUTER SYSTEM

CIRCLE NO. 1 ON INQUIRY CARD

☐ **Enclosed is $395 for my SwTPC Computer Kit** ☐ **Send Data**

☐ **or BAC**________________ **#**__________

☐ **or MC**________________ **Ex Date**__________

NAME

ADDRESS

CITY STATE ZIP

Southwest Technical Products Corp., Box 32040, San Antonio, Texas 78284

SCCS INTERFACE

The National Publication of the Southern California Computer Society

Vol. 1 Issue 4 **MARCH 1976**

COVER STORY .... Pg. 9

THE SOUTHERN CALIFORNIA COMPUTER SOCIETY

OFFICERS

Ward Spaniol	President
Louis G. Fields	Vice President
Don Tarbell	Secretary
Hal Lashlee	Treasurer

BOARD OF DIRECTORS

Pat Amornvitikivacha–'75-76
Art Childs–'75-'76
Louis G. Fields–'75-'76
Roy Fulton–'75-'76
Chris King–'75-'76
Marvin Perlman–'75-'76
Ron Rogers–'75-'76
Jerry Silver–'75-'76
Don Tarbell–'75-'76
Carl Todd–'75-'76
Bob Baskin–'75-76
Ray DeLong–'75-'76
John French–'75-'76
Lois Heiser–'75-'76
Ruben Loshak–'75-'76
Larry Press–'75-'76
George Rowe–'75-'76
Mike Stern–'75-'76
George Tate–'75-'76
Pearce Young–'75-'76

OBJECTIVES AND PUBLICATIONS

A non-profit organization whose purpose is:

A. To exchange and disseminate information among the Society members concerning the computer arts and sciences.

B. To provide a technical assistance to other members of the Society in those computer projects which are not undertaken for pecuniary gain or profit including but not limited to hardware, software, and computer programming.

C. To publish books, newsletters, magazines, and other periodicals for the benefit and education of the Society members and the general public.

D. To control and sponsor seminars, lectures, and courses relating to the computer arts and sciences.

F. To develop and maintain computer centers and laboratory workshops for the members of the Society and the general public including provisions for time-sharing operations.

SCCS INTERFACE, the official publication of the Southern California Computer Society is published monthly. Its content is composed primarily of articles contributed by members of the Society and intended for the reader with an interest in computers for professional or avocational reasons.

MEMBERSHIP

Membership in the Society is open to anyone, regardless of educational background or geographic location. All members are entitled to the rights of meetings, elections, receipt of official publications in a timely manner, and all other benefits as provided by the officers, Board of Directors, and various committees of the Society pursuant to the by-laws of the Southern California Computer Society.

MEMBERSHIP CLASSIFIED ADVERTISING

Each month SCCS Interface will devote some space free, non-commercial advertising by members. This is done as a service to members of the Southern California Computer Society and to help promote communications between computer experimenters and hobbyists throughout the world.

Many SCCS members, being pioneers in the small computer field, often have for sale, on an occasional basis, used computing equipment, or equipment they have designed and built themselves. Those members, as well as those who simply wish to establish contact with individuals with similar or complimentary interests are encouraged to utilize the member ad privilege subject to the following conditions:

A. Ads must be received by the ad editor of SCCS Interface by the first of the month prior to issue month.

B. Ads may list the member's name, home address and phone. No commercial firm names, please!

C. Ads must be typewritten, double spaced and no more than 250 characters (including spaces and punctuation) in length. Ads 250-500 characters will be printed on space-available basis.

D. Ads may be run up to three consecutive issues, however, a specific request must be received for each instance.

E. No free member ads will be accepted for the sale of equipment or services by a commercial profit-oriented individual or organization.

Address all member ads to: Interface Ad Editor, Nancy DeLong, 8005 Denrock Ave., Los Angeles, Calif. 90045

Those non-members wishing to advertise are invited to join SCCS and enjoy the member ad privilege as well as the delivery of SCCS INTERFACE to your mail box.

CONTENTS

Features

Introduction to Microprocessor Technology 10
Reprint Courtesy: Motorola Semiconductors, Phoenix, Arizona

Teleprinter Maintenance/Repair/Preventative Maintenance 18
By Cliff Sparks

Serial Communication with Software 21
By Terry Benson, Field Applications Engineer, Intel Corporation, Santa Clara, Ca.

Calculatin' Engines 26
By William E. Peterson

The Raw and the Cooked 29
By John Whitney

Featurettes

Join the Club? 36
By Ward Spaniol

Software Documentation Standards 46
By Bill Roch

Buzz Words for the Computer Enthusiast 49
By Terry Benson, Field Applications Engineer, Intel Corp., Santa Clara, Ca.

Departments

Applications Exchange 38
Bits & Bytes 39
Book Review 34
Games and Things 40
Interfacial 3
Letters to the Editor 44
New Products 32
Update 6
Unclassified Ads 59

SCCS EDITORIAL STAFF

Art Childs, Editor
Sheila Clarke, Editorial Assistant
Scott Wilcox, Hardware Editor
Jeff Landry, Software Editor
Nancy DeLong, Associate Editor
Mike Stern and Tom Rugg, Games Co-Editors
Mike Teener, Bits 'n' Bytes Editor
Larry Press, Applications Exchange Editor
Cliff Sparks, Peripherals Editor
Bob Martin, Photography
Membership Circulation, John French

EDITORIAL CONTRIBUTIONS

They must be accompanied by return postage and we assume no responsibility for loss or damage thereto. Any material accepted is subject to such revision as is necessary in our sole discretion to meet the requirements of this publication. The act of mailing a manuscript and/or material shall constitute an express warranty by the contributor that the material is original and in no way an infringement upon the rights of others. Send all contributions to: Art Childs—Editor, 335 North Adams, Suite 210, Glendale, Ca. 91206.

PUBLISHERS STATEMENT

All editorial content and effort are contributed at no cost to the Society. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the Southern California Computer Society. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes an endorsement of the product or products by this magazine or by the Southern California Computer Society.

Interface magazine is published monthly by McPheters, Wolfe & Jones, 6515 Sunset Blvd., Suite 202, Hollywood, Calif. 90028. Subscription rate $1.00 with membership in Southern California Computer Society.

Application to mail at second-class postage rates is pending at Los Angeles, California 90051.

POSTMASTER: Please send change of address form 3579 and undelivered copies to: Southern California Computer Society, P.O. Box 3123, Los Angeles, CA. 90051.

DIRECT ALL ADVERTISING INQUIRIES TO: Bruce Berkey, National Sales Manager, Advertising Department, SCCS INTERFACE, 61 South Lake Avenue., P.O. Box 4566, Pasadena, California 91106—(213) 681-0685.

The One and Only

Altair® 4K Static.

Altair 4K Static from MITS is unquestionably the finest 4K static memory available anywhere. It is also the fastest.

Altair 4K Static uses Intel 2102 A-4 memory chips which have a worst case access of 450 nanoseconds at 70 °C. At normal system temperatures the access times are typically less than 300 nanoseconds.

Altair 4K Static is fully isolated from the system bus by Schmitt ™ Triggers. Thus, the excessive capacitive loading caused by other 4K static memories is eliminated. Use of these triggers on all Altair 4K static inputs greatly reduces noise. Internal data collection nodes also use Schmitt Triggers, which prevents internal data bus noise from being transmitted to the system data bus.

Altair 4K Static is the only 4K static supported by MITS. Owners of Altair 4K Static are eligible to qualify for discounts on AltairBASIC and other MITS products.

Altair 4K Static is the only 4K static that comes with all the required Altair hardware including edge connectors and card guides.

Altair 4K Static is the answer for Altair owners who need static memory for special applications such as the TV Dazzler from Cromemco.

PRICES:

Altair 4K Static Kit	**$159**
Altair 4K Static Kit with 2K Memory	**$134**
Chip set to convert 2K to 4K	**$ 45**

SPECIAL—Altair Documentation Notebook. Contains catalog, price sheet, Computer Notes newspaper, Software Information Package, technical data on Altair hardware, list of authorized Altair dealers, list of computer clubs, survey of home computing market, and much more. All in top quality three ring binder. Only $5 plus $1 for postage and handling. Offer expires April 30, 1976.

MAIL THIS COUPON TODAY

Enclosed is check for $ __________
BankAmericard # __________ or Master Charge # __________
☐ Altair 4K Static ☐ Altair 2K Static
Include $3 for postage and handling
☐ Please send me the Altair Documentation Notebook. Enclosed is $5 plus $1 for postage and handling.
☐ Please send free catalog and price sheet.

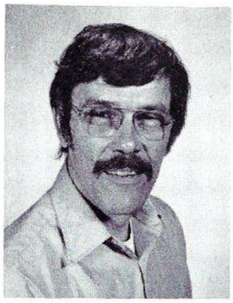
NAME __________
ADDRESS __________
CITY __________ STATE & ZIP __________

MITS/2450 Alamo SE/Albuquerque, NM 87106/505-262-1951

Prices, delivery and specifications subject to change.

mits
2450 Alamo SE/Albuquerque, NM 87106/505-262-1951

CIRCLE NO. 2 ON INQUIRY CARD



Art Childs, Editor

Photo by George R. Clarke

INTERFACIAL

When I assumed editorship of INTERFACE a few short months ago, I naively wrote the following editorial. Until now, I'd felt the time wasn't right for posing such a question. And now it's too late . . . with members in Hong Kong as well as most of the United States, and requests coming in from places like India and Holland. It is obvious to me that SCCS has been handed a new problem. Before I go into that—let's look at the earlier problem and my initial thoughts.

MICRO OR MACRO?

Should SCCS become a large, perhaps national organization such as IEEE has become, or should we remain small, limited strictly to Southern California . . . even the Los Angeles area?

Judging from discussions at two general meetings and several meetings of the Board of Directors, feelings on this question are divided. Many wish the club to remain essentially small and advocate a "town meeting" format. There may be merit to such an approach for our organization, if only to avoid the "bigness" that seems to have engulfed us as members of a modern society.

Those who would have us avoid becoming too big argue that it is essential to maintain an atmosphere conducive to face-to-face encounter, implying that such is the best form of communication, the best way to exchange ideas and information, the most comfortable way to relate to our contemporaries. One might tend to agree before considering the advantages of a large organization.

Southern California seems to be one of those areas in this country where the new things are happening. Why? Climate? Chance? The kind of people who are attracted to the southwest? There seem to be more doers, innovators and creators who prefer living in Southern California than any other comparable area in the country. One result of this is the kind of interest in computers that has caused a group of a few people to grow in five months to a group of over five hundred with only word of mouth to communicate its existence.

Southern California however, has no monopoly on the kind of talent and interest that is evident in our club. There must be thousands of high school students, college students, and working people with intense interest in computers who, because of geographic location, find themselves isolated from the kinds of activities and organized interests we as members of SCCS enjoy. Shouldn't those people also be invited to share as members?

There are many other reasons for not limiting the size or geographic extent of our organization, among them the advantage of greater resources; i.e., better computer center(s), *a more comprehensive and interactive publication (INTERFACE), and of course, the most important—more human creativity.*

The proposition put forth for your consideration is: wouldn't it be preferable to have the best of both worlds, that is, a large, perhaps national organization with all attendant benefits, and still maintain, via interest groups and local chapters, the highly desireable quality of informality that accompanies face-to-face communication? Why not let fellow members know how you feel through INTERFACE.

It seems like light years (and a few Klingons) ago since that was written.

The new problem can best be illustrated by Carl Helmers' statement in the February issue of BYTE Magazine. Carl, editor of that fine publication wrote:

"All the reasons which make a computer society a useful item to have around one's own back yard also argue for an organization which is national in scope. Is it time for such an organization in this area of small scale computing? Most likely yes: There is already indication that some form of nationwide affiliation is beginning to form.

I refer to that contradiction in terms, the *Delaware Valley* (i.e., Philadelphia area) chapter of the *Southern California* Computer Society, which is mentioned in the Clubs and Newsletters section of this issue of BYTE. (It would seem strange and somehow provincial for an organization aspiring to nationwide scope to retain the name Southern California Computer Society.) The idea is right, even if an appropriate name is not yet available. From the start, the people who have been running the Southern California Computer Society have been doing an excellent job with the programs which make up an active organization of this sort. This evaluation is based upon the SCCS*Interface* newsletter and upon phone conversations with individuals who are involved with SCCS."

I must admit that besides feeling good about the complimentary statement (and a bit bewildered by the use of the term "newsletter"), I was struck by the fact that the name of our organization certainly is a contradiction in terms. The structure of SCCS will probably become tri-level in nature. Besides the formality and business of a national (or international) organization, we should see the formation of many local chapters and, within the chapters—or quite possibly across chapter boundaries—the gradual emergence of special interest groups. Should such turn out to be the case, a change of name will certainly be necessary.

So, what do we call ourselves? The Benevelant Binary Brotherhood? or the Great Intergalactic Computer Society? (Mike Stern and the Klingons certainly deserve recognition) or perhaps the Sactimonious Corroborating Computer Socialists (an effort to preserve our SCCS logo is in order so we won't have to change our art work). In any case, it seems we're soon due for a new handle. What's your preference?

IN THIS ISSUE

The software feature this month is the second in a series by Terry Benson, in which he discusses the fine details of timing with an 8080 microprocessor. Terry also contributed a set of buzz words to help the beginners get into the technical conversation.

John Whitney discusses more ideas about computers and graphics and to point the programming effort in the right direction, Bob Williams reports on a book on the subject.

Bill Peterson laments the passing of the slide rule with some interesting history in "Calculatin' Engines", and Ward Spaniol suggests, in "Join the Club?" that our current name, SCCS, might also become history.

Larry Press offers some suggestions on writing tutorials for INTERFACE in his column, and Mike Teener takes time off from the slopes to keep our hardware appetites whetted—as usual.

Tom Rugg has joined the staff to assist Mike Stern with "Games and Things", and for his first contribution, discusses the all-important random number generator.

In the Hardware Department, Scott Wilcox discusses some interesting aspects of protected memory, and other goodies. For those of us who prefer to think in hex rather than octal, Dick Kenyon offers a simple but effective fix for the front panel of the Altair.

Standards—that all-important subject that must be dealt with sooner or later—is the subject broached by Bill Roch in "Software Documentation Standards". Bill has offered to act as a clearing house for ideas and discussion on the subject. Serious home computerists are invited to write him, or INTERFACE, and help get the standardization effort off to a good start.

Our education in microprocessor theory is continued in this issue with another installment of the Motorola series, and in the mysteries of TTY with more from Cliff Sparks.

Id' like to pose a question regarding the future content of INTERFACE. What would you like to see—in what areas would you like more effort concentrated? I am hoping eventually to gather a staff large enough to do a thorough job of educating and writing on any aspect of home computing. To do this, I will need the cooperation of many people with talent and expertise to share, and some direction from the readership. More writers to serve dual functions of editor and referee are needed—particularly from areas other than Los Angeles. We in Southern California sincerely want to know what's going on in other parts of the country (world?) so if you's care to contribute rather than edit, we'll welcome you to our pages. Put that last statement in spades if you have 6800 software contributions. We're having a difficult time locating anything for the Sphere and MITS 680 owners.

See you in Albuquerque, byte-bitten brothers, where I'm looking forward to trading the TTL for a well pickled olive, viewing the latest MITS contribution to our crazy hobby, and anticipating the pleasure of meeting you in person.

Last, but not least, an editorial by yours truly concerning an area I've heard a lot of griping about, but unfortunately, have yet to see any real effort applied toward the solution needed. The editorial concerns the quality of equipment, parts and service we hobbyists receive from those organizations to whom we send our money. I hadn't intended to attempt a project of this nature until summer, but frankly, I'm more than a little discouraged about the treatment some members are receiving. At the same time, the respect shown for the consumer by some manufacturers is laudable. Both the good and the bad deserve public (computing public) attention. The questionaire on page ( ) is designed to aid in bringing to the attention of all future purchasers of home computing equipment the past performance of those manufacturers and supliers with whom members have done business. If you, the reader, will take time to read the editorial and fill out the questionaire (make as many copies as you require) it will help us at INTERFACE to make life for you, the purchaser, less frustrating in the future.

Speaking of the future, I'd like to pose a question regarding the future content of INTERFACE. What would you like to see—in what areas would you like more effort concentrated? I am hoping eventually to gather a staff large enough to do a thorough job of educating and writing on any aspect of home computing. To do this, I will need the cooperation of many people with talent and expertise to share, and some direction from the readership. More writers to serve dual

(*Continued on page 48*)

Solder less. Design more.

The soldering iron. As a design tool, it's obsolete.

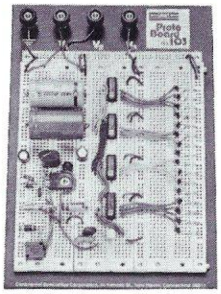
Whether you're a full-time engineer or spare-time hobbyist, there are only so many hours a day you can spend designing and building circuits. So why not make the most of your time?

Design circuits literally as fast as you can think. Instead of starting your next project with a soldering iron, save time by starting with our Design Mate™ 1. In a compact (7½"W x 6¾"D x 3¼"H) case with a convenient sloping top, it offers solderless breadboarding at its best: 790 terminals; a continuously-adjustable regulated 5-15 VDC 600 mA supply; and a DC voltmeter to monitor the internal supply or test your operating circuit.

With Design Mate 1, hooking up (or changing) a circuit is as simple as pushing leads into holes on the breadboard. Rugged 5-point contacts insure reliable, low-resistance connections between resistors, capacitors, transistors . . . even IC's in TO-5 or DIP packages. And short lengths of solid #22 AWG wire make interconnections easy wherever you need them.

At $49.95* complete, it's not only a time-saver, it's a money-saver, too.

Larger capacity? Smaller capacity? Have it your way. Whatever your breadboarding needs, we can fill them. For larger capacity, save time with our Proto-Board® series—660 to 3060 terminal points, with or without built-in regulated power supplies. For smaller applications, use 1 or more of our handy QT Sockets and Bus Strips, locking them together for infinite expandability, as you need it. Starting at $2.00.



Proto-Board 103. 2250 solderless terminals save you time on every circuit you design.

Test DIP's with power on and hands off—even on the most crowded circuit board with our Proto-Clip™ series.

Other ways to save time and money. Proto-Boards, QT Sockets and Bus Strips are just the beginning of the many ways we can help make your life more enjoyable: how about a foolproof, shortproof way to test DIP IC's—even in operating circuits—for just $4.50*? A full-featured, precision function generator for $64.95*? A professional R/C bridge for $54.95*? A way to instantly see and monitor logic levels in an operating IC, without expensive scopes or elaborate procedures. And more, to save you time, money and aggravation.

Design Mate 2: versatile function generator, only $64.95*

Design Mate 3: Precision R/C bridge, just $54.95*

Time is money: Start saving it. Ask your CSC dealer to show you how to solder less, design more and take the drudgery out of your electronic life. Or send for our free catalog and dealer list.

*Manufacturer's recommended retail.

CONTINENTAL SPECIALTIES CORPORATION

CSC

44 Kendall Street, Box 1942
New Haven, CT 06509 • 203-624-3103 TWX: 710-465-1227
West Coast office: Box 7809, San Francisco, CA
94119 • 415-421-8872 TWX: 910-372-7992
Canada: Len Finkler Ltd., Ontario

© 1975 Continental Specialties Corp.

CIRCLE NO. 3 ON INQUIRY CARD

UPdate

MARCH MEETING - Saturday, March 20th
TRW Systems, Redondo Beach
Building "S" Cafeteria

Doors open 10 a.m. Coffee hour is 12 - 1 p.m. Meeting 1-4:30 p.m. Refreshments are available at coffee hour, and to sustain the kitty, a $1 is requested.

APRIL MEETING - Saturday, April 24th.

The JANUARY MEETING enjoyed an attendance of over 500 and more than 50 new members signed up. The meetings are beginning to look more and more like a "computer swap meet", with members bringing their own items to sell ranging from "how to" books to I/O devices, to completely assembled minis. The increasing number of exhibitors is giving the meeting a "bazaar" look. Add this to the very informative talks made by Jonathan Titus, co-author of "The Bug Books", and member Jerry Silver who gave us a broad base for programming languages, and we all got a lot to see, hear and discuss. Door prizes only increased an already frenzied atmosphere, and some lucky fellows walked away with Bug Books 1, 2 and 3 donated by Edwards Associates, VTVOR EICO given by E. Schubart, RCA'S Data books and logic templates, Calculator and parts cabinet donated by SCCS, and the biggie, a Vector power wire wrap tool donated by the Computer Mart. Thank you TRW . . . we couldn't do it without you. And additional thanks to member Ron Keele for putting it all together.

SCCS UPCOMING EVENTS

Altair Convention . . . Everyone's talking about what they'll be exhibiting and entering into the CONTEST. It almost looks like an SCCS event, so be reminded to take care of your flight and room reservations through Majestic Travel (213) 657-0441. WACC dates are March 26-28.

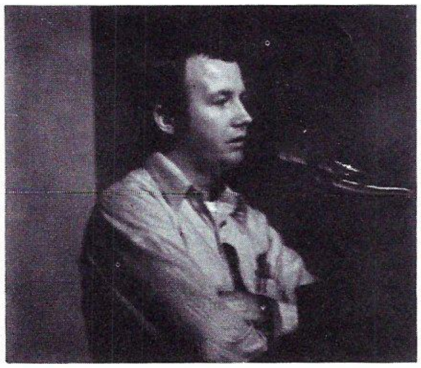
SCCS SPECIAL MEETINGS are announced via telephone taped message. To stay updated, call (213) 472-0388.

TWO NEW CLASSES TO BE OFFERED

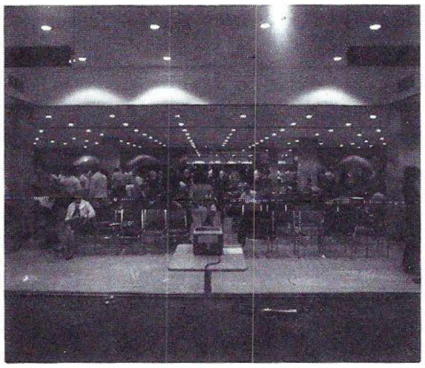
The SCCS education program is underway. By the time you read this, our first hardware construction project—an 8080 based CPU—will be finished. It was a terrific learning experience for those of us involved in it and future issues of INTERFACE will carry articles describing in detail what we did for the benefit of those who didn't make the class. We have nearly finished our first BASIC classes. Nearly 40 people now speak BASIC to their friends and computers, thanks to Jerry Silver and Gene Morrow.

Now that we have finished our CPU it's time to start a second hardware design and construction workshop. This time Pat Amornvitikivacha will lead a project in which a Tarbell compatable (look who is becoming a quasi-defacto standard) *cassette interface* is designed and constructed. Pat estimates that the class will run for four sessions with some study in between. The class will meet on Sundays and run roughly from 10 a.m. to 2 p.m. Tuition will be $15, however, you will save more than that on the construction of your cassette interface (apologies to Don Tarbell). Even more important, you will learn a good deal about circuit design and construction techniques because Pat is an excellent teacher and a very clever fellow. This class is not for the absolute beginner—you should have some basic knowledge of electronics, gates, and logical design, but nothing too heavy is required.

We will also begin an *APL class* during March. The class will be led by Marv Perlman, who teaches and



Coauthor of "Bug Books" Jonathan Titus speaks to membership at SCCS January meeting about range of far reaching possibilities for applications in home computing.
Photo by Bob Martin



SCCS January meeting at TRW Systems Cafeteria attracts over 500 members and interested newcomers. Included are member and manufacturer exhibitors, swappers, exchanging information and wares.
Photo by Bob Martin

JERRY SILVER addreses SCCS January meeting on programming languages.
Photo by Bob Martin

uses APL at the Jet Propulsion Laboratories. The class will meet for 8 or 10 Saturday afternoon sessions at the Harvard School in North Hollywood. Tuition will be $25 to cover machine time. The test is roughly $8 more. There are no prerequisites for this class.

If you wish to enroll in either class, send your tuition (make checks to SCCS) to Larry Press, 128 Park Place, Venice, CA 90291 or call (213) 399-2083 for further information. By the way, if you find the tuition a bit steep for you right now, we have a *scholarship fund* to help you out.

Finally, let us know if you would like to organize a class or project. See the February INTERFACE for prior information on offering classes.

RESOURCE COMMITTEE TO PUBLISH CALENDAR

The Resource Committee, headed by Ruben Loshak, is attempting to coordinate all SCCS events. Ruben welcomes your suggestions for new activities, and invites announcements of upcoming events through the Resource Committee. Volunteers, equipment and locations for events are also being solicited. Your note which includes the name of the activity, the sponsor group, date, time and place can be sent to Ruben at 4040 Badillo Circle, Apt. 50, Baldwin Park, Calif. 90706, and the information will get on the calendar which is planned to go out to all members.

A HOME AT LAST FOR SCCS!

The Harvard School in North Hollywood has generously provided SCCS with an unused dormitory to be converted to office space. After considering a number of possibilities SCCS Board of Directors voted unanimously to accept the offer of complimentary facilities. The Harvard School personnel are fine people to work with and are very interested in our projects. By the time you read this, we expect to have begun formation of a library, workshop, and office. By April we'll be able to publish our official new phone number and address.

Obviously, much organization will be required to make it all work, and it should come as no surprise that we could use your help setting things up. How about it, you collectors of desks, chairs, tables and file cabinets?—can you donate or loan your unused items that you've been planning to gather for your next garage sale? For those stacks of books and magazines that are now too high to even guess what's at the bottom . . . we'll gladly help you make space for more. Hardware, shelving, lighting and volunteer labor are all needed.

Members have already stepped forward to coordinate the many aspects of putting our facilities in shape. For library contributions, contact Mike Dent (213) 677-5433 or Allison Martin (213) 329-8941. The workshop is being planned by Lou Fields (213) 272-0942 and Mike Dent. If you need more information or can help in any way, call Larry Press (213) 399-2083 or Art Childs (213) 243-5179.

If there's a place to sit and something to sit on, we'll be able to hold committee meetings, work on joint projects, assemble a library, hold classes, and have an information center . . . the endless possibilities boggle the mind! We'll be looking for your contributions and offers of help to make SCCS headquarters a viable work center.

SCCS ACTIVITIES

The San Fernando Valley Chapter is in full swing. The next meeting is scheduled at the Harvard School on Wednesday, March 10th at 7:15 p.m. For information, contact John Scott at 2435 N. Lamer St., Burbank 91504.

EAST COUNTIES CHAPTER is now forming for the Orange County area. Interested members should contact Ron Rogers (714) 523-2417. Hobbyists who haven't yet made up their minds to become members because of distance may find the Orange location very convenient.

SAN GABRIEL VALLEY CHAPTER ORGANIZING . . A chapter of SCCS for members living in and about the San Gabriel Valley is now being formed. A final organizational meeting, open to all interested parties will be held late in March. Temporary officers will be elected and chapter goals will be discussed. The exact time and location of the meeting will be announced at the SCCS meeting on March 21st. Details may also be obtained by writing Dan Erickson, at 400 S. Catalina Ave., Pasadena 91106. Please include a stamped self addressed envelope for your reply.

VENTURA GROUP MAY CHAPTERIZE . . . Approximately 40 computer buffs attended the first meeting of the Ventura County group to take those first steps toward becoming an organized club. The attendees, whose experience with computers range from novice to expert, and from hardware through ham radio to software, considered several issues including the desireability of becoming an SCCS Chapter. The enthusiasm typical of computer club meetings was enhanced by the display of two Altairs up and running, one which sported a very useful modification—a switch to cause the machine to "step" through a program at approximately 10 instructions per second. John Borders, the innovator, has promised to supply the details for publication. Thanks John, and good luck to the Ventura group.

SCCS has its very own computer doctor. George Tate, one of the SCCS "founding fathers" has decided the computer population is great enough to warrent his full time attention. We trust his malpractice insurance rates will never be excessive, and wish him success in his new business venture. You can find him at (213) 663-2604 when he's not making house calls.

CLUB NEWS

The Computer Organization of Los Angeles (COLA) an education oriented computer group is sponsoring their First Annual Computer Conference on March 6th. The purpose is to disseminate information on computer applications in education and will focus on the interests of teachers, administrators and computer specialists. The conference will take place at LeConte Jr. High School in Hollywood. For more information, contact COLA president, Alyce Jackson (213) 292-8237, or write her at P.O. Box 43677, Los Angeles 90043.

HAVE YOU READ MICRO-8?

WE SUGGEST YOU CONSIDER DOING SO. The Micro-8 Computer User Group Newsletter, edited and produced by John Craig and Hal Singer, is packed with letters from readers living in every part of the country as well as many notes of general interest to the home computerist.

Advertising seems to be almost non-existent, the comments from the readers appear to be real, and the editors obviously are sincere in their effort to serve the serious computer hobbyist. NL, as it is referred to, tells it like it is—no punches pulled. For example, one reader writes, "SCCS (*Interface*) will soon find they cannot say things that need to be said. It is extremely difficult to criticize any manufacturer or supplier when he is, or potentially is, an advertiser."

We don't necessarily agree with that writer, but do agree that he has the right to state his opinion and respect Craig and Singer for giving him—and it appears, anyone else—the opportunity to say it.

Be our guest and copy the subscription form . . . find out for yourself. Send it care of Cabrillo Computer Center, 4350 Constellation Rd., Lompoc, Calif. 93436.

SUBSCRIPTION FORM

☐ .... Volume 1 Back Issues 1 thru 4 $3.50

☐ .... Volume 1 Back Issues 5 thru 12 6.00

☐ .... Volume 2 Back Issues 1 thru 6 6.00

Name ____________________

Address ____________________

____

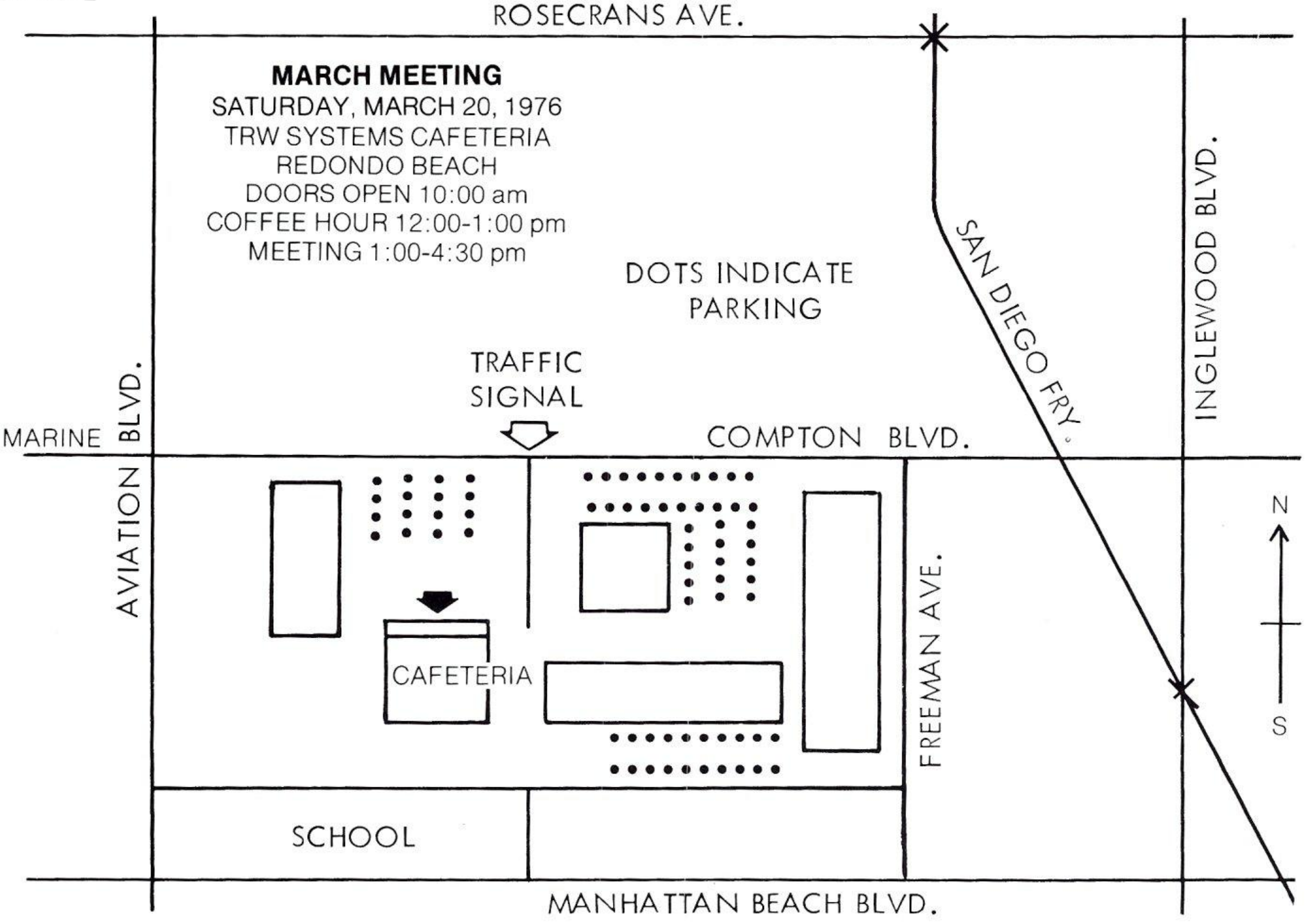
Zip ____________________

Telephone No. ____________________

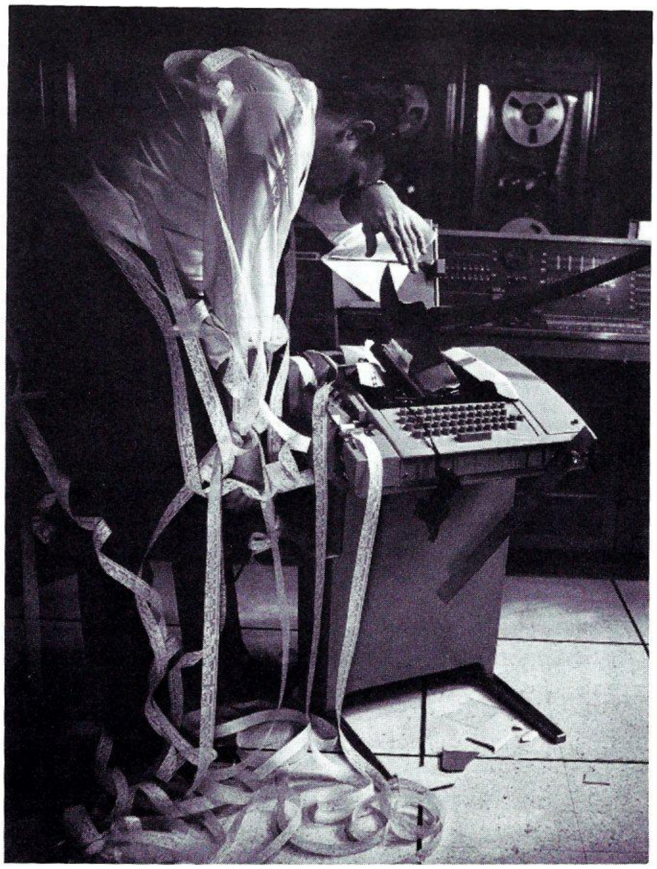
(May be published — leave blank if you prefer)

Also include (if you desire) a little note or letter describing your equipment, plans for the future, experience, etc.

Thank you



MARCH MEETING
SATURDAY, MARCH 20, 1976
TRW SYSTEMS CAFETERIA
REDONDO BEACH
DOORS OPEN 10:00 am
COFFEE HOUR 12:00-1:00 pm
MEETING 1:00-4:30 pm



Serving Southern and Northern California

Rents • Sells • Leases
Services new and used
Terminal Equipment

Call TSI for Teletypes, Decwriters,
CRT's of all types,
Modems and Couplers

Featuring
Datamedia Elite™ CRT Terminals

Call Today
for the Best in Service

(213) 769-6772

TERMINAL SYSTEMS INC.
11300 HARTLAND AVENUE NORTH HOLLYWOOD CALIFORNIA 91605

CIRCLE NO. 4 ON INQUIRY CARD

COVER STORY

"MOMENT OF NEED"

Our friend pictured here is exhausted, frustrated and bereaved from a fit of jealous rage. He found that his loyal and most trusted terminal had been out the night before with another computer, at a byte swapping party and has lost its unblemished keys to unscrupulous hands. He decided to commit the ultimate act of terminal communion in his weakest moment! (Don't axe him why.)

Terminal Systems Inc., of North Hollywood insists that one of their service men captured this moment with his Polaroid just before the straight-jacket team arrived for this poor fellow.

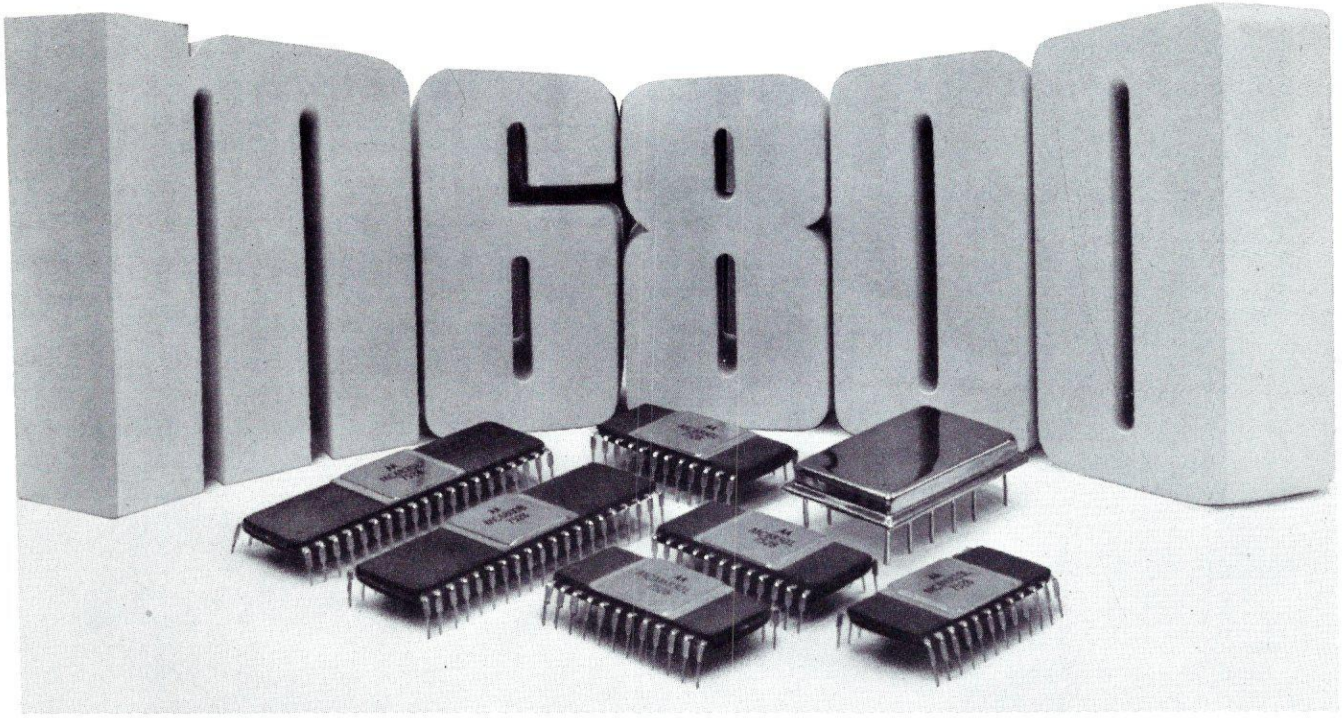
In addition, he had almost made a terminal case out of three Decwriters, two CRTs, one acoustic coupler, and a modem in a pear tree. Sorry, but the folks at Terminal Systems Inc. do not have spare parts for pear trees!

They suggest you avoid terminal situations by leasing or buying from TSI when your moment of need arises—Better yet—Call them *before* it becomes terminal!

No warranty is offered on axe repairs!

Reprints of the cover are available from Terminal Systems, Inc., 11300 Hartland St., North Hollywood, CA. 91605, or call (213) 769-6772.

INTERFACE is indebted to Terminal Systems Inc. for providing this cover picture, as all can relate who have touched those delicate keys and caressed the 1/0 cables and felt the anguish of terminal frustration.



Second part

Application to microprocessors

Chapter III

USE OF THE STACK

We have seen in the previous chapter that the stack could be used for the temporary storage in sequential order of the data contained in the two accumulators.
We will now describe two more frequent uses of the stack.

1. Use of the stack in calling a subroutine

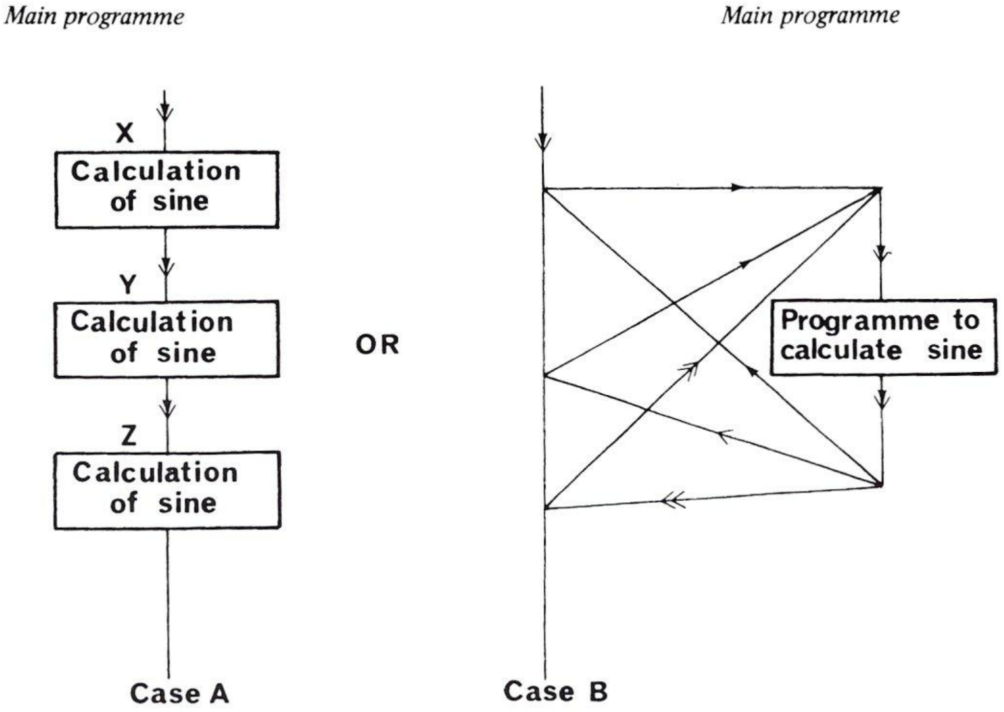
1.1. *Definition of a subroutine*

A subroutine or sub-programme is a programme performing a given function likely to be used frequently by the main programme.
Example:
Suppose that in some main programme the sine of a number has to be calculated when the PC reaches the values X, Y, Z, that is, when the instructions located at addresses X, Y, and Z are to be carried out; this can be accomplished schematically as follows:

INTRODUCTION TO MICROPROCESSOR TECHNOLOGY

From "Introduction to Microprocessors" reprint courtesy of Motorola Semiconductor, Phoenix, Arizona.

Interface has serialized this new and very informative book and will be presenting it in its entirety over the next several issues.


Main programme
Main programme
X
Calculation of sine
Y
Calculation of sine
Z
Calculation of sine
OR
Programme to calculate sine
Case A
Case B

- in case A we see that the programme for the calculation of the sine, which makes use of series expansion ($\sin x = x - \frac{x^3}{3!} - \frac{x^5}{5!} - \ldots$) and is rather long and complex, must be written in 3 different places in the programme; assuming that it occupies N positions in the memory, it is clear that 3N positions will be necessary to programme the calculation of the sine at 3 points of the main programme;
- this is why it is more usual to use the second method (case B) in which the programme for the calculation of the sine, written once only, is called each time it is required. In our example, the method leads to a saving of about 2N memory positions.

1.2. *Calling a subroutine*

Calling a subroutine SUBR is done by means of the instruction JSR SUBR (Jump to Subroutine) in all cases or by the instruction BSR SUBR (Branch to Subroutine) only if the symbolic address SUBR is within -125_{10} or $+129_{10}$ positions of the address of the instruction BSR.

1.3. *Return to the main programme*

The last instruction of the subroutine must be the instruction RTS (Return from Subroutine) which allows a return to the main programme and execution of the instruction which would have been processed if this sub-programme had not been called; this instruction is held at the address termed "return address".

1.4. *Example:*

The respective sines of two numbers with addresses X_1 and X_2 are to be calculated and the results stored at the addresses Y_1, Y_2. We will call for the sub-programme SINE which can, for example, calculate the sine of a number contained in the accumulator A (ACCA) and leave the result in the accumulator.

The essentials of the programme are as follows:

```
load (X1) in ACCA
calculate SINE (X1)
store ACCA at the address Y1
load (X2) in ACCA
calculate SINE (X2)
store ACCA at the address Y2
stop.
```

This programme is written as follows:

```
Address  |  Instruction
add1     |    LOAD A X1
add2     |    JSR SINE
add3     |    STORE A Y1
add4     |    LOAD A X2
add5     |    JSR SINE
add6     |    STORE A Y2
add7     |    HALT
add8     |  SINE_┌ _ _ _
         |       ¦
add9     |    RTS
```

Note:
The addresses of the instructions must not be regarded as separated by one unit; the instruction JSR for example is coded on 3 bytes so that add3 (address 3) = add2 + 3.

1.5. *Description*

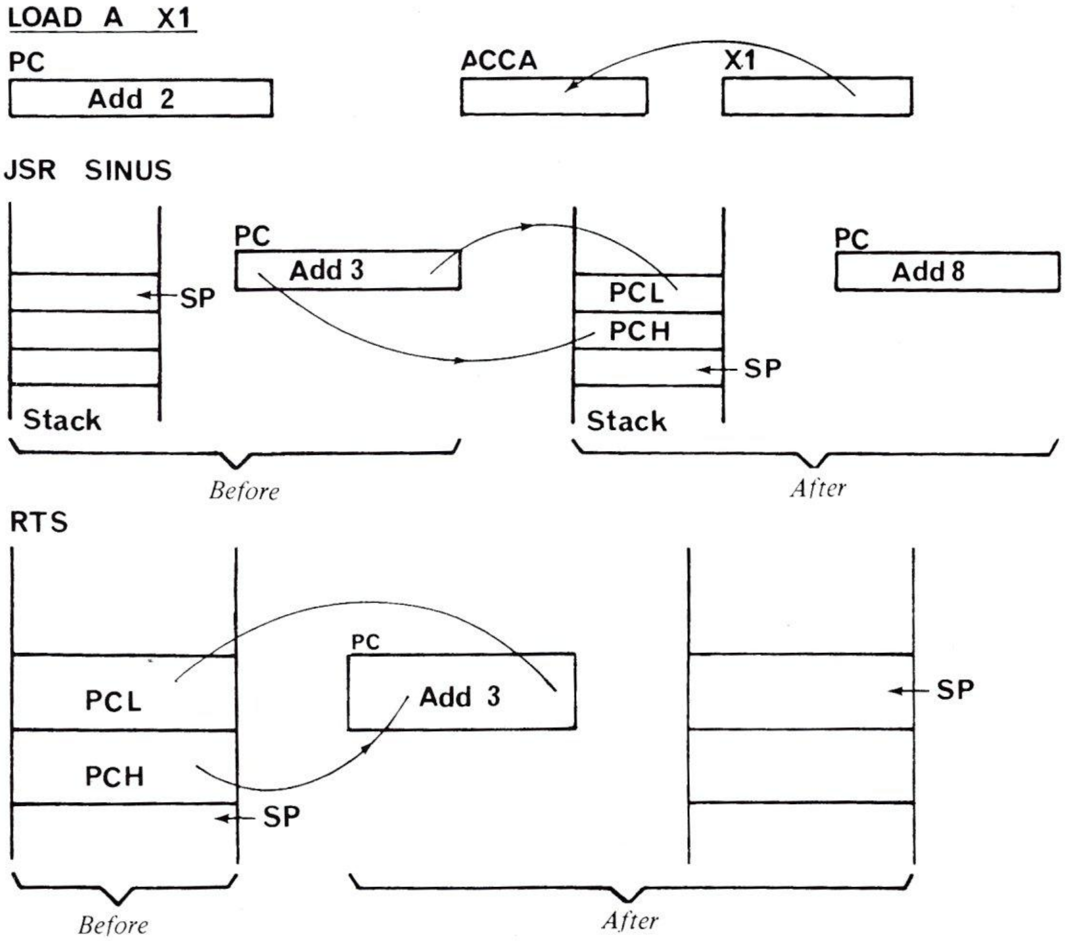
After execution of the instruction LOAD A X 1, the PC (Programme Counter) indicates the address add2. The CPU will therefore, in a first phase, fetch the instruction JSR SINE and increment the PC which will then contain the value add3. In a second phase, the instruction JSR SINE consist of two automatic operations:

a) Saving the return address add3 contained in the PC: the contents of the PC (16 bits) are pushed into the stack (2 bytes)

b) Branching to the symbolic address SINE: the PC is loaded to the value add8.

The instructions stored from add8 onwards are then executed until the instruction RTS is encountered. When the instruction RTS is executed, the PC is loaded to the value add3 which is pulled from the stack; the instruction stored at the address add3 will then be executed, i.e., STORE A Y1, etc.

The different operations may be represented as follows:


LOAD A X1
PC
Add 2
ACCA
X1
JSR SINUS
PC
Add 3
SP
Stack
PCL
PCH
SP
Stack
PC
Add 8
Before
After
RTS
PCL
PCH
SP
PC
Add 3
SP
Before
After

1.6. *Nesting of subroutines*

We have just considered the case where a single sub-programme is called, in this case, the sub-programme SINE. Very often, a subroutine SUB1 calls a subroutine SUB2 which in turn may call a subroutine SUB3, etc.

The MPU MC6800 permits the accomplishment of such "nested" subroutines to an infinite level, as the processing of the sub-programme uses the stack, which may be of infinite size. In contrast, other micro-processors keep the return address in one of the internal registers instead of in a zone of the memory. The level of nesting is then limited to the number of these stack pointer registers.

2. Use of the stack for an interruption

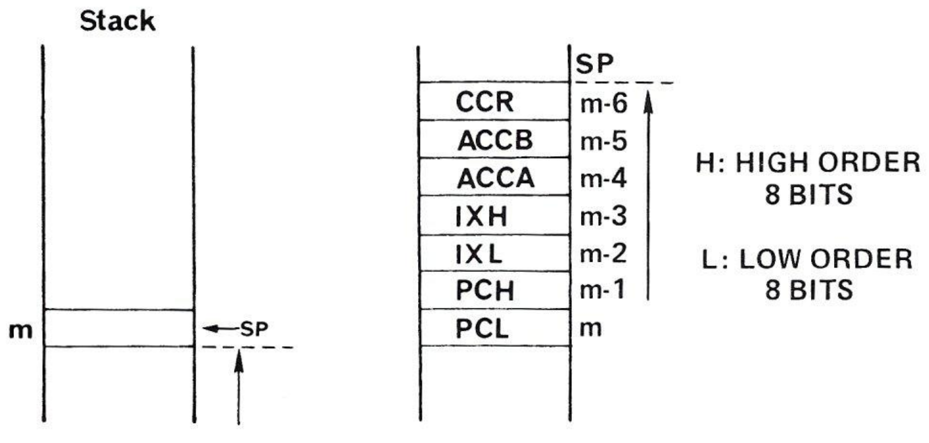
The processor can be interrupted and forced to execute an interrupt service routine. An interruption can have one of the following four origins:

- restart;
- powerfail;
- programmed interrupt SWI (Software Interrupt);
- interrupt request by a peripheral.

In the case of the last three possibilities, there is automatic dumping on the stack of the complete state of the system; the state of the system is completely defined by:

- the contents of each of the registers ACCA and ACCB;
- the contents of the status or condition code register CCR;
- the contents of the index register IX;
- the contents of the Program Counter which will allow the CPU to execute the instructions left aside as a consequence of the interrupt request.

Automatic dumping takes place in the following order:



Before *After*

The dumping of the state of the system constitutes the first phase in the execution of an interruption.

In a second phase, the Processor will execute the interrupt routine starting with the address indicated by the PC.

An interrupt routine generally ends by an instruction RTI equivalent to the instruction RTS used in the subroutines in that it provokes the pulling of the values dumped on the stack to their respective destinations; in the case of an interrupt it is the complete state of the system which is re-established, that is, the values defining the state of the system just before the interruption.
In particular, the PC is reloaded with the address of the instruction which would have had to be executed if there had been no interruption; the execution of the interrupted programme can thus be continued.

2.1. *The four cases giving rise to an interruption*

2.1.1. *Restart*

When the system is switched on, or on resetting, the PC is loaded with the value contained in the last two bytes, the highest in the memory. It is thus possible to execute a routine termed resetting.

2.1.2. *Powerfail*

This is the opposite case to the previous one in that the interruption is generated by the mains voltage dropping to an abnormally low value. The external system must have provision for detecting the drop in mains voltage and generating an interruption which will allow the state of the MPU to be dumped in a non-volatile memory. In this case, the interrupt vector, that is, the two addresses containing the value to which the PC must be loaded, are located at the addresses n-2 and n-3, n designating the highest numerical address.

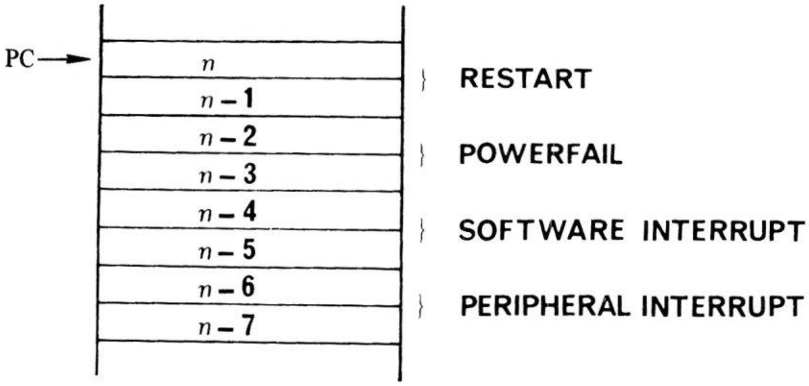
2.1.3. *Software interrupt SWI*

The interruption is generated by the programme and not by hardware. The vector is located at the addresses n-4 and n-5. This instruction SWI finds its utility in the "step by step" execution of a programme, that is, instruction by instruction; the EXBUG programme used in the EXORCISER is based on this instruction.

2.1.4. *Peripheral Interrupt*

The MPU can be interrupted by a peripheral device via a PIA or an ACIA (Asynchronous Communications Interface Adapter). Provided the Interrupt Mask bit in the Condition Code Register is not set, the MPU will first save its registers onto the stack and then branch to the peripheral interrupt service routine, the address of which will be fetched from locations n-6 and n-7. The WAI instruction (Wait for Interrupt) enables to serve the interrupt request as quickly as possible by anticipating the storage of the MPU registers on the stack. This will save at least 8 cycles.

2.1.5. *Recapitulation of the interrupt vectors*

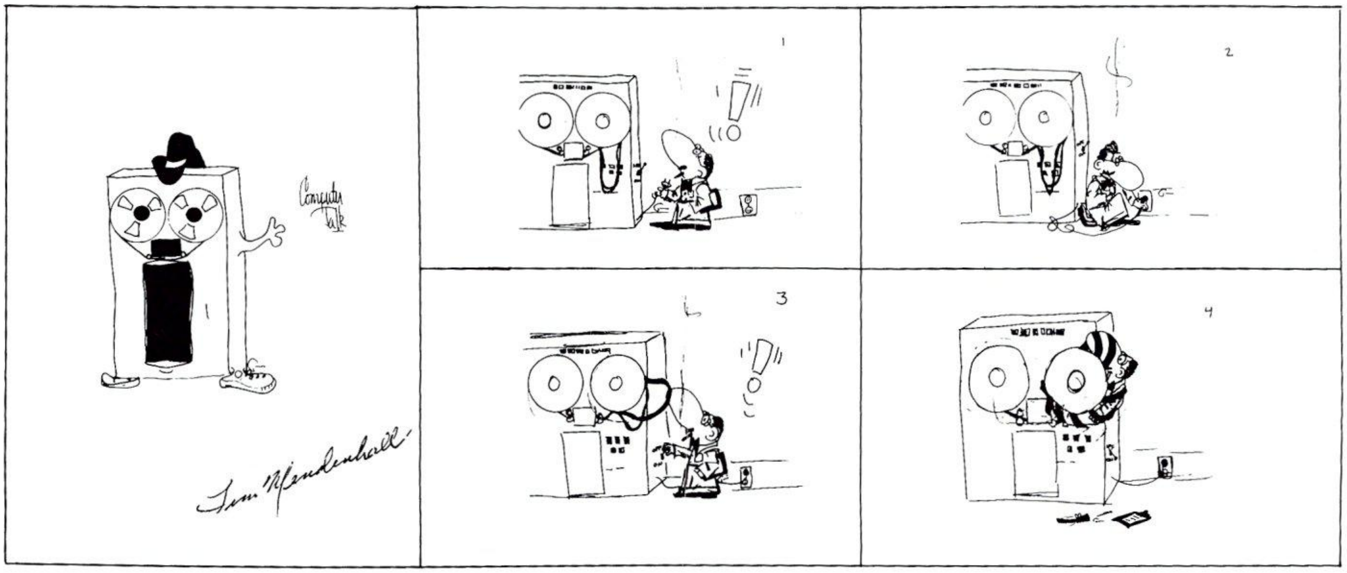

PC
n
n – 1
n – 2
n – 3
n – 4
n – 5
n – 6
n – 7
RESTART
POWERFAIL
SOFTWARE INTERRUPT
PERIPHERAL INTERRUPT

2.1.6. *Interrupt nesting*

In the same way as it is possible to have a structure of nested subroutines, nested interrupt routines may be envisaged.

For this, it is sufficient to remember (see Part I) that a peripheral can interrupt the MPU while the latter is executing an interrupt routine provided that the interrupt request is not masked (bit I of the status at 0). It will be remembered that this allows different priorities to be assigned to the service routines of the peripherals.

(Continued on page 52)


Computer Talk
Tom Mendenhall
1
2
3
4

MOM AND POP HEISER:

The Nation's First and Foremost Computer Dealers

Still today offering the best in microcomputers and service.

- MITS Altair™ computers and peripherals
- Southwest Technical TV Typewriter II
- Cromemco TV Dazzler™ and Bytesaver™
- Polymorphic Video Display and D/A
- Professional Hardware and Software Assistance

Drop by and soak up some of the homey atmosphere at

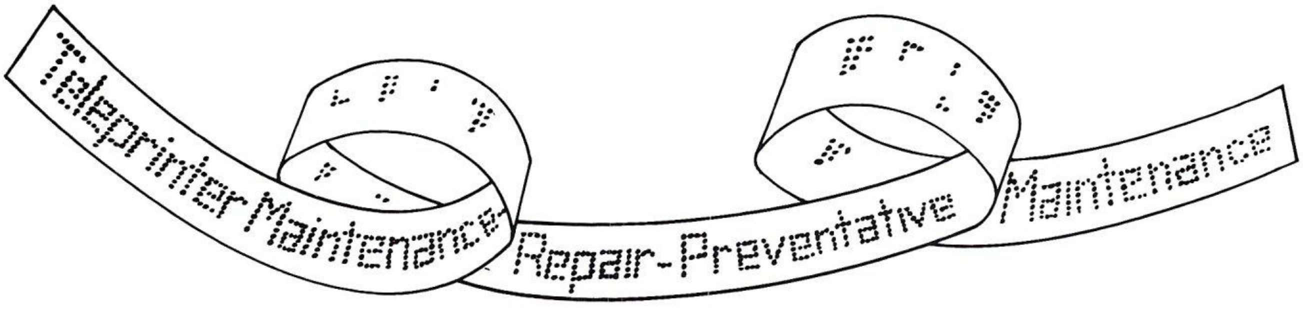
THE COMPUTER STORE

Arrowhead Computer Co.
11656 W. Pico Blvd.
Los Angeles, CA 90064

2-8 p.m. Wed.-Fri.
10 a.m.-6 p.m. Sat./Sun.

(213) 478-3168
BankAmericard/Master Charge
Welcome

CIRCLE NO. 5 ON INQUIRY CARD



by Cliff Sparks

This is the fourth article in a series describing the operational functions of the 33 ASR/KSR.

In our third article we were introduced to the basic functions of the distributor clutch and the operation of the main shaft clutch assembly. We again wish to thank the TELETYPE CORPORATION, 5555 Touhy Ave, Skokie, Illinois for the photo's that will be displayed in this article.

There are four clutches in the Model 33 ASR/KSR. The internal operation of each clutch is the same, however its external use may differ. The clutch is made up of two halves, one consisting of the clutch drum, the other its clutch disc. Sandwiched between the two halves are the clutch latchlever, shoes and spring assemblies.

The clutch shoes, springs and latchlever are assembled to the clutch disc. The clutch drum (when engaged) rotates with the entire assembly. When disengaged the drum rotates while the clutch disc may remain stationary.

The first of the four Model 33 clutches is the distributor clutch, then the codebar clutch, function clutch and finally the selector clutch.

Lets review our basic rule on clutch operation. If a given clutch rotates in a clock-wise direction, then pressure against the clutch latchlever in a counter clock-wise direction will cause that clutch to disengage.

Clutches are tripped by the clutch trip lever and locked by the seating of the clutch latch lever. Refer to Figure 1.

Clutch disengagement really means that any mechanism attached to the disc side (or half) of the assembly will not be activated.

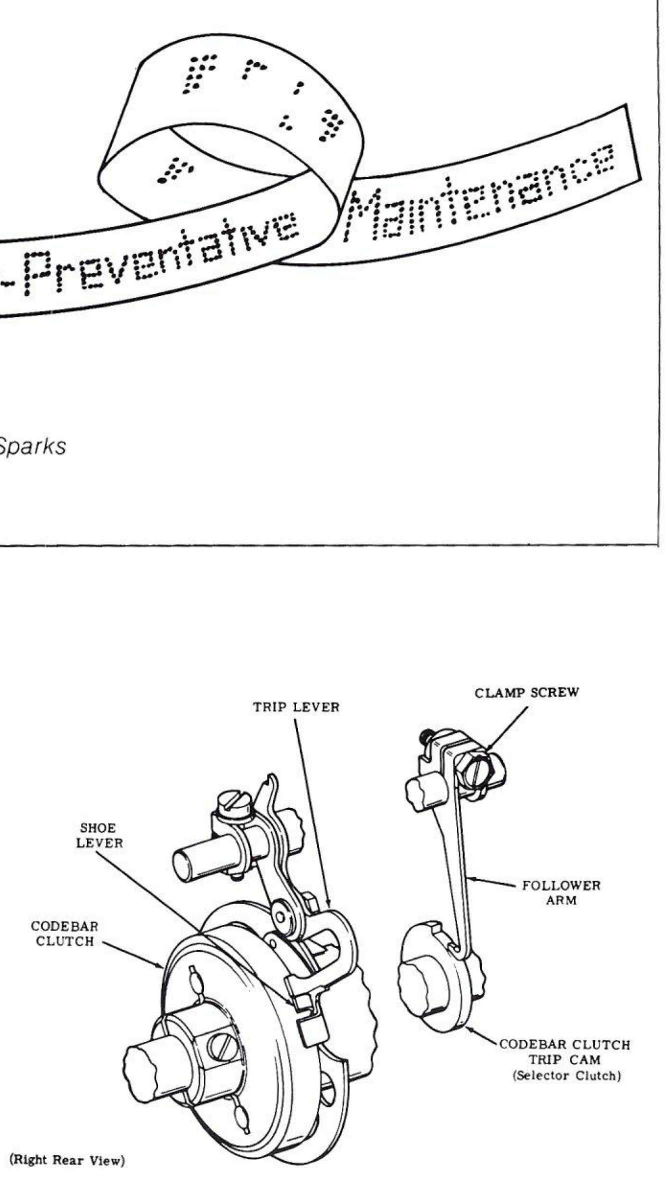


Figure 1

Therefore, if our clutch is latched and the clutch trip lever has not been energized the clutch latchlever will remain in the unattracted position.

If we allow ourselves to fall into the area of over simplifying a detailed mechanical operation, then so be it—. Having explained the clutch operation, and understanding that these clutches are always mounted on a shaft, it is simple to generalize and state the following:

a) The cam side of the clutch (clutch disc) will most often have a caming surface attached to it. Its operation will cause action to occur at a precise time.

b) Clutch engagement may be timed to allow the clutch to engage or disengage when required. Refer to figure 1a.

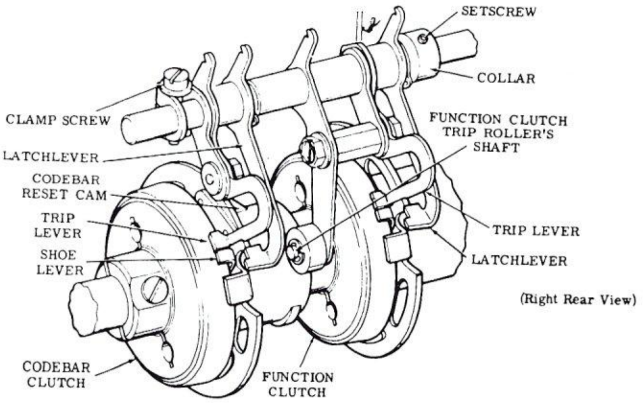

SETSCREW
COLLAR
CLAMP SCREW
FUNCTION CLUTCH TRIP ROLLER'S SHAFT
LATCHLEVER
CODEBAR RESET CAM
TRIP LEVER
TRIP LEVER
SHOE LEVER
LATCHLEVER
(Right Rear View)
CODEBAR CLUTCH
FUNCTION CLUTCH

Figure 1A

To observe our clutch in operation refer to figure 2. Note that the upper right corner of the illustration shows the distributor clutch and its shaft, but fails to indicate that on the other end of the distributor shaft is the Brush Holder. The holder is sweeping a pair of contacts over the face of the segment disc. The distributor shaft is mounted thru a hole provided in the center of the distributor disc.

The distributor gear pulley, driven by the motor, motor gear and motor belt cause the distributor clutch drum and the drive gear, bolted to it, to rotate. This rotation drives a gear mounted on the end of the main shaft.

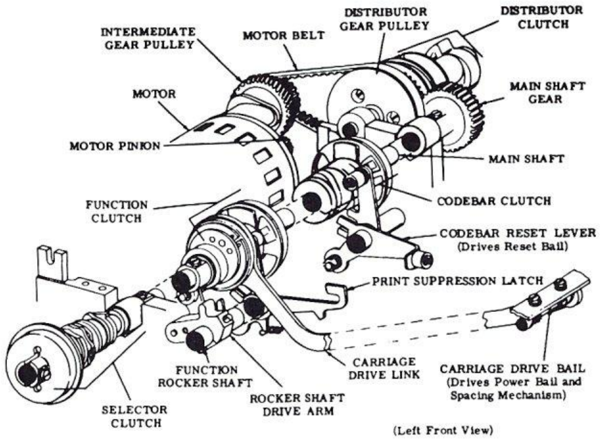

INTERMEDIATE GEAR PULLEY
MOTOR BELT
DISTRIBUTOR GEAR PULLEY
DISTRIBUTOR CLUTCH
MOTOR
MAIN SHAFT GEAR
MOTOR PINION
MAIN SHAFT
FUNCTION CLUTCH
CODEBAR CLUTCH
CODEBAR RESET LEVER (Drives Reset Bail)
PRINT SUPPRESSION LATCH
FUNCTION ROCKER SHAFT
CARRIAGE DRIVE LINK
CARRIAGE DRIVE BAIL (Drives Power Bail and Spacing Mechanism)
ROCKER SHAFT DRIVE ARM
SELECTOR CLUTCH
(Left Front View)

Figure 2

It is safe to assume that when power is applied to the machine, the motor will cause the main shaft to rotate. If the clutches are properly adjusted and a character has not been selected, the following events will occur.

1. Main shaft will rotate.
2. Codebar clutch will remain disengaged.
3. Codebar reset lever will remain undisturbed, since the caming surface on the codebar clutch has not been allowed to rotate.
4. Function clutch will remain disengaged. The function clutch cam will remain undisturbed.
5. The selector clutch assembly, consisting of the Selector cam and clutch assembly will remain undisturbed. The selector clutch drum will rotate with the main shaft, not the screw head on the collar of the clutch drum. It is securing the drum to the main shaft. All the clutches are assembled in a similiar manner. Refer to Figure 3.

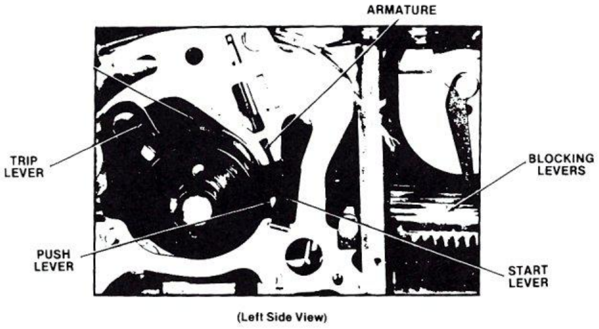

ARMATURE
TRIP LEVER
BLOCKING LEVERS
PUSH LEVER
START LEVER
(Left Side View)

Figure 3

Figure 4 is a view taken from the rear of the machine. It clearly shows the motor at the bottom of the photo, the distributor clutch to the left, its pulley, and the main shaft drive gear. Across the center of the picture is the main shaft. At this point I want to assure the reader that both the codebar clutch and function clutch are shown, but difficult to see.

Note the bar and shaft mounted over the main shaft. Mounted to the bar are the clutch trip levers (the hook like devices) and the clutch latch lever (shown just below the trip levers). These levers are pivoted from their mounting shaft. They are held under spring tension by springs attached to one end of the lever, and the other to a bar mounted between the bar supports. The entire assembly as described, is referred to as the trip shaft assembly.

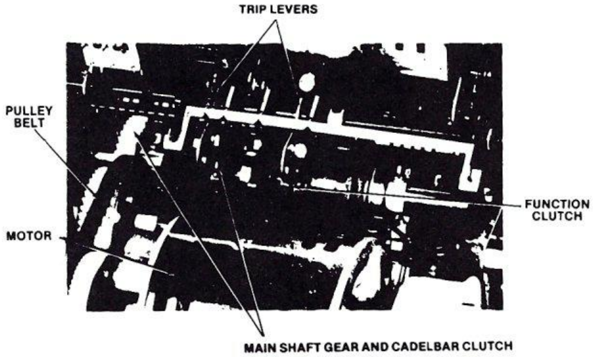

TRIP LEVERS
PULLEY BELT
FUNCTION CLUTCH
MOTOR
MAIN SHAFT GEAR AND CADELBAR CLUTCH

Figure 4

Figure 5 will require the use of your imagination, but will illustrate that the trip shaft shown has a trip follower arm bolted to one end of it. The entire shaft is mounted thru a bearing like surface at both ends. This allows the entire shaft to rotate a few degrees when

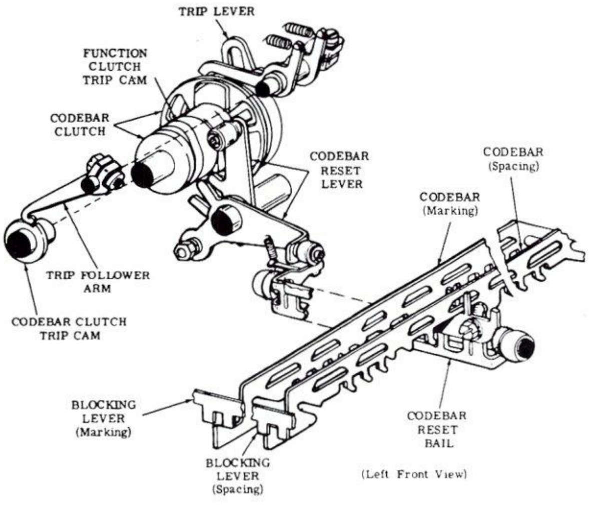

TRIP LEVER
FUNCTION CLUTCH TRIP CAM
CODEBAR CLUTCH
CODEBAR RESET LEVER
CODEBAR (Spacing)
CODEBAR (Marking)
TRIP FOLLOWER ARM
CODEBAR CLUTCH TRIP CAM
BLOCKING LEVER (Marking)
BLOCKING LEVER (Spacing)
CODEBAR RESET BAIL
(Left Front View)

Figure 5

required. Therefore, if the clutch trip levers and trip follower arm are bolted securely to the trip shaft, any movement of the shaft will lift not only the trip follower arm but the codebar clutch trip lever, and cause the associated codebar clutch to engage.

Figure 6 shows the assembly as described. Please note the follower arm resting on the caming surface of the selector cam.

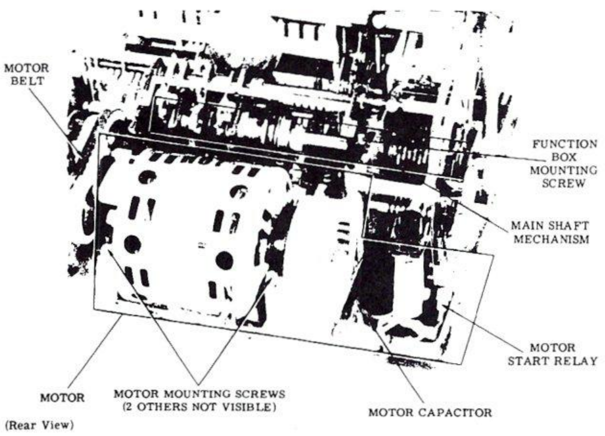
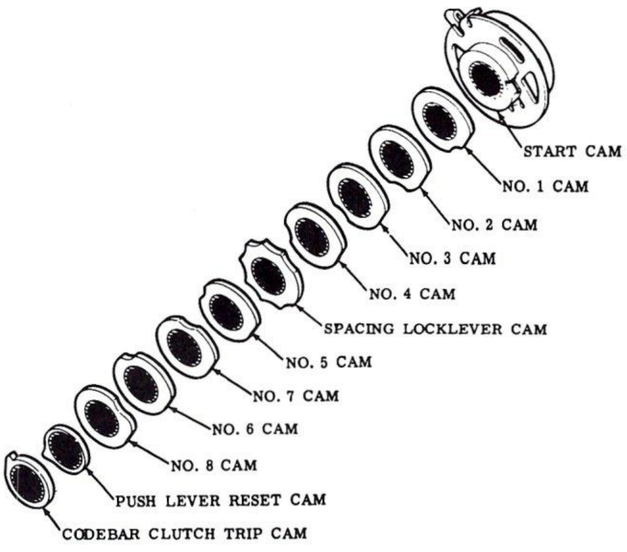

MOTOR BELT
FUNCTION BOX MOUNTING SCREW
MAIN SHAFT MECHANISM
MOTOR START RELAY
MOTOR
MOTOR MOUNTING SCREWS (2 OTHERS NOT VISIBLE)
MOTOR CAPACITOR
(Rear View)

Figure 6

When the selector cam rotates, the caming action causes the trip follower arm to be raised to the high lobe of the codebar clutch trip cam, further rotation of the cam will cause an abrupt drop in the follower arm.

Figure 7 shows the cams as if they were removed from their shaft. This photo allows the reader to better comprehend the unique shape of each cam. It must be understood that the cam lobes are fused together and rotate as a unit.

The rotation of the selector cam is being held in check by the selector clutch. Its disengagement allows the main shaft to rotate while the cam itself remains stationary.

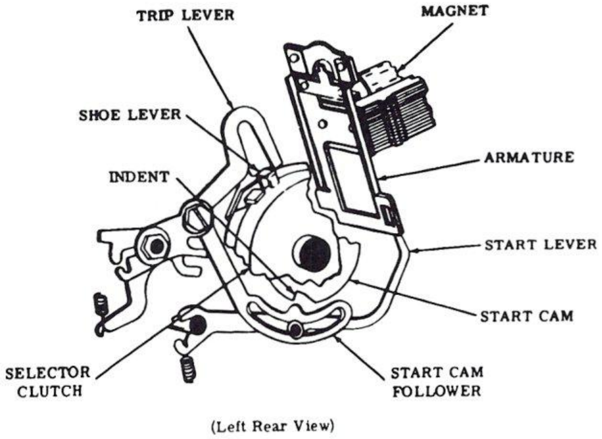

START CAM
NO. 1 CAM
NO. 2 CAM
NO. 3 CAM
NO. 4 CAM
SPACING LOCKLEVER CAM
NO. 5 CAM
NO. 7 CAM
NO. 6 CAM
NO. 8 CAM
PUSH LEVER RESET CAM
CODEBAR CLUTCH TRIP CAM

Selector Cam Sleeve

Figure 7

At this point lets review a bit.

1. Our machine has power applied.
2. The motor and main shaft are rotating.
3. Four disengaged clutches are preventing the following actions from occuring:
   a. Distributor brush holder from sweeping carbon brushs over distributor disc.
   b. Codebar clutch prevents its caming surface from rotating.
   c. Function clutch prevents it caming surface from rotating.
   d. Selector clutch prevents its selector cam from rotating.


TRIP LEVER
MAGNET
SHOE LEVER
INDENT
ARMATURE
START LEVER
START CAM
SELECTOR CLUTCH
START CAM FOLLOWER
(Left Rear View)

Selector Trip Mechanism

Figure 8

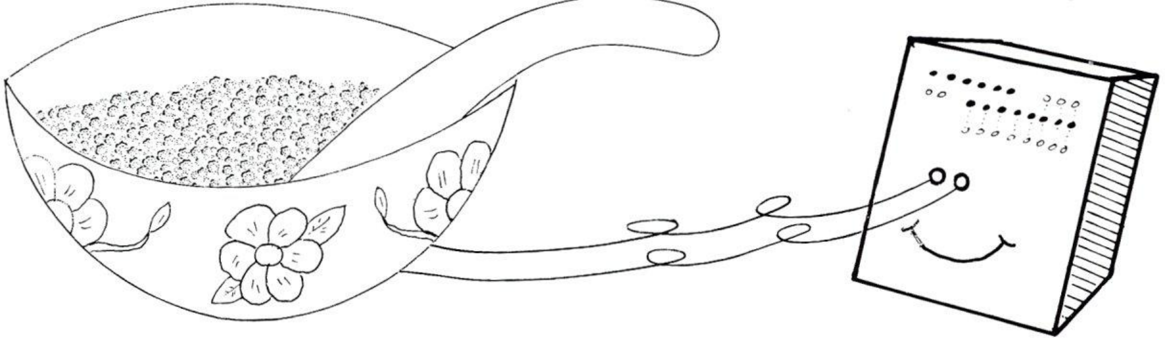
Figure 8 shows us the selector clutch, shoe lever, trip lever and their relationship to the selector armature. The mechanism receives the code combinations from the selector magnet driver and

(Continued on page 50)

Microcomputer Applications

Serial Communications with Software

by Terry Benson
Field Applications Engineer, Intel Corp.



In the previous article, we discussed the timing capabilities of microcomputers with an example of a timing loop for the Intel 8080. The program that we developed is shown in Figure 1 so that the timing calculations can be computed. In the program for generating a pulse there are four sections that must be included to calculate the total cycle time:

1) Set-up, D1;
2) pulse generation, C2;
3) main timing loop, C3;
4) exit loop (or zero loop), C4.

The total time required for program execution is:

T= C x .5 usec. (for 8080 using 2 MHZ clock.)

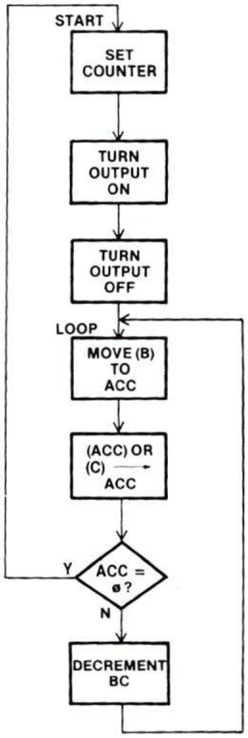
Where T is total time in seconds and C is the total number of clock periods: C = C1 + C2 + N x C3 + C4, where N = value of COUNT. From line 1 the set-up (C1) requires 10 cycles. From lines 2 through 5, pulse generation (C2) required 32 cycles. The main timing loop (C3), if N is non-zero, requires 34 cycles per loop. If N is zero, then there are only 19 cycles executed (C4). Thus the total number of cycles is:

```
Line
 No.   Cycles  Label    Operation        Comments
  1      10    START:   LXI B, COUNT     ;LOAD BC WITH COUNT VALUE
  2       7             MVI A, 2         ;USE BIT 1 (00000010)
  3      10             OUT 8            ;OUTPUT CONTENTS OF A TO PORT 8
  4       5             XRA A            ;CLEAR A
  5      10             OUT 8            ;OUTPUT TO PORT 8
  6       5    LOOP:    MOV A, B         ;GET HIGH ORDER 8 BITS
  7       4             ORA C            ;OR WITH LOW ORDER 8 BITS
  8      10             JZ START         ;IF ZERO, GO TO NEXT PULSE
  9       5             DCX B            ;DECREMENT 16 BIT COUNTER
 10      10             JMP LOOP         ;GO CHECK FOR ZERO
```

PROGRAM TO GENERATE A PULSE
FIGURE 1

But N must be an integer, thus N will be set to 266. With this value, the actual time period will be 4.5525 msec which is only 2.5 usec larger than specified. (Notice that the resolution in time periods for this example is 17 usec.)

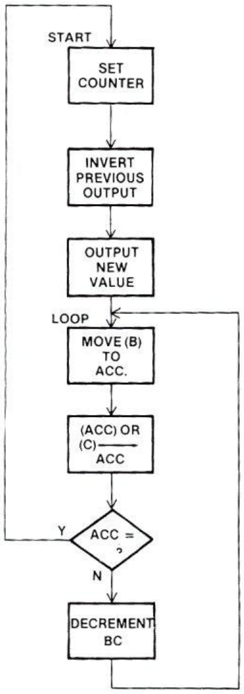
The second part of your "independent research" was to modify the program such that it will generate a square wave output, i.e. with a 50% duty cycle. Let's look at the flow chart first to decide how to effect the easiest change. Figure 2 represents the original program for generation of a pulse. By simply



Flow Chart to Generate a Pulse

Figure 2

converting the two blocks that generate the pulse to two blocks that invert the previous data, we have the needed program (see figure 3). This means, however, that the data going to the output port must be accessible by the program so that the bit can be inverted.



Flow Chart to Generate Square Wave
Figure 3

The following set of instructions illustrate a method for saving the output data and modifying only one bit. The memory location can be referred to by a label called IMAGE since it is the image of the output port.

Line No.	Cycles	Label	Operation	Comments
1	10		LXI H, IMAGE	;SET MEMORY POINTER
2	10	START:	LXI B, COUNT	;LOAD BC WITH COUNT VALUE
3	7		MOV A, M	;GET PORT IMAGE
4	7		XRI 2	;INVERT BIT 1 (00000010)
5	7		MOV M, A	;SAVE PORT IMAGE
6	10		OUT 8	;OUTPUT TO PORT 8
7	5	LOOP:	MOV A, B	;GET HIGH ORDER 8 BITS
8	4		ORA C	;OR WITH LOW ORDER 8 BITS
9	10		JZ START	;IF, ZERO, GO TO NEXT PULSE
10	5		DCX B	;DECREMENT 16 BIT COUNTER
11	10		JMP LOOP	;GO CHECK FOR ZERO

PROGRAM FOR GENERATING SQUARE WAVE
FIGURE 4

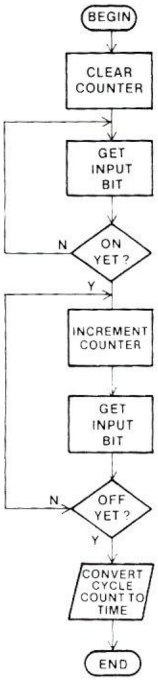
This set of instructions allows the output port data to be stored into a memory location so that selected bits can be modified without disturbing the other bits. This is one method for using all eight bits of an output port while each bit controls a different function. See figure 4 for the complete program for generating a square wave signal.

With the program operating in this mode note that the loop time will represent only half of a total cycle with two passes through the program required to complete a full period.

TIMING AN EXTERNAL EVENT

With the knowledge of the timing characteristics of microcomputers, a program can also be written that will time an external event. A program can be used to time the duration of an event—such as in a switch debounce application where the switch input must be on for a certain period of time before it is considered closed. Another type of timing program might be used to determine the time between two events—such as in determining the speed of a motor where the time between revolutions is used to calculate the RPM.

We can develop an example of a program that times an external event. In order to use this type of program, some restrictions must be placed on the input to be evaluated. First of all the pulse must always be wider than the program loop time. Secondly, the pulse cannot exceed the program capacity. If only 16 bits are used, the pulse time must be shorter than that which would overflow the 16 bit counter. The flow chart of figure 5 illustrates the two logical functions to be performed: 1) check for a true signal to start, and 2) increment a counter while checking for a false signal. Figure 6 shows a program that can be used for timing the event.



Timing an External Event
Figure 5

In this program the minimum pulse width that can always be detected must last longer than from line 2 to 4—or 27 clock cycles.This means that for guaranteed detection the input pulse must be wider than 13.5 usec for an 8080 operating at 2MHz. The maximum pulse width can also be determined. The number of cycles in loop 2 (L2) is 32 which means that each time through the loop will represent 16 usec. With a sixteen bit counter, the maximum pulse width of just over one second. So this program can measure an event of from 14 microseconds to 1 second. (The CNVRT routine is not included.)

I will leave it as an exercise for you to modify this program so that it can be used to measure the time between two events.

Line No.	Cycles	Label	Operation	Comments
1	10	BEGIN:	LXI B,O	;CLEAR REG BC FOR COUNTER
2	10	L1:	IN 3	;GET ALL 8 BITS FROM PORT 3
3	7		ANI 4	;CHECK BIT 1 (00000100)
4	10		JZ L1	;IF NO SIGNAL, CHECK AGAIN
5	5	L2:	INX B	;INCREMENT COUNTER
6	10		IN 3	;GET INPUT SIGNAL
7	7		ANI 4	;MASK OFF OTHER 7 BITS
8	10		JNZ L2	;IF STILL ON, INC COUNTER
9	17		CALL CNVRT	;CONVERT VALUE TO TIME

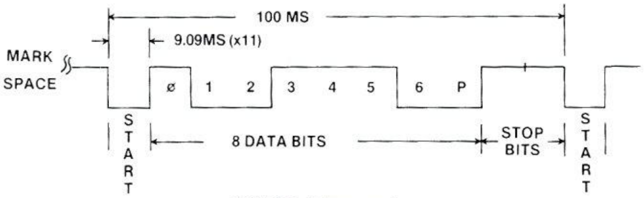
PROGRAM FOR TIMING EXTERNAL EVENT
FIGURE 6

SERIAL DATA COMMUNICATION

Another use for the timing of external events is in the area of serial data communication. To reduce the amount of external circuitry in a microcomputer system, a program can be used to convert parallel data to a serial bit stream and a serial input stream to parallel data. Most of the keyboard/display or keyboard/printer terminals employ a serial data communication scheme and it is likely that your microcomputer system will use one of these devices. (The primary reason for using the serial transmission technique is that only about one fifth the number of wires are required to connect two devices together. This is especialy beneficial when communicating over telephone lines.) The most commonly used serial communication format is the eight level, eleven bit ASCII (American National Standard Code for Information Interchange) coding.

In order to illustrate the use of a microcomputer communicating with a teleprinter we will first have to discuss the serial data format. The most popular device, as you may have guessed, is the Teletype. For this reason the remainder of the discussion will be directed toward Teletype communication at an asynchronous data rate of 110 baud (basically 110 bits per second). Naturally the microcomputer can comunicate with a terminal operating at higher baud rates and the program techniques derived here can be modified to accommodate other devices.

The fact that data is asynchronous means that it is not "synchronized" with a clock source. Each word is independent of all other words and timing is related only to the bits within that word. Each word contains 8 data bits and 3 "overhead" bits. The first bit—start bit—is used to indicate that a word is beginning. The next 8 bits are the data bits (one of which is normally the parity bit) and are delayed from the start bit by a fixed amount. The last two bits—stop bits—are used to separate two words and allow a certain degree of timing tolerance in the data bit stream. After a minimum time, equivalent to 2 bit times, another start bit can occur. Figure 7 illustrates the timing relationships with the number "9" being transmitted. A mark, or 1, is the high state and a space, or zero, is a low state. (For teletype, 20 ma. current is marking, no current is a space.) The least significant bit is transmitted first—the last bit, P, is the parity bit.



ASCII Format
Figure 7

The actual bit time is determined from the transmission rate. At 110 baud, the maximum *bit* rate is 110 bits per second. (Remember this rate is the rate of data bits as well as the start and stop bits so that the actual word rate is only 10 words per second.) At 110 bits per second, each bit time will be 1/110 or 9.09 milliseconds.

Using the timing techniques previously described, we can write a program that generates a serial data stream. The eight bits of data must be serialized with a time delay subroutine invoked after each bit is output. The steps required to accomplish this serialization will be developed here and are illustrated in flow chart form in figure 8. Figure 8a shows the data serialization while figure 8b shows the delay subroutine.

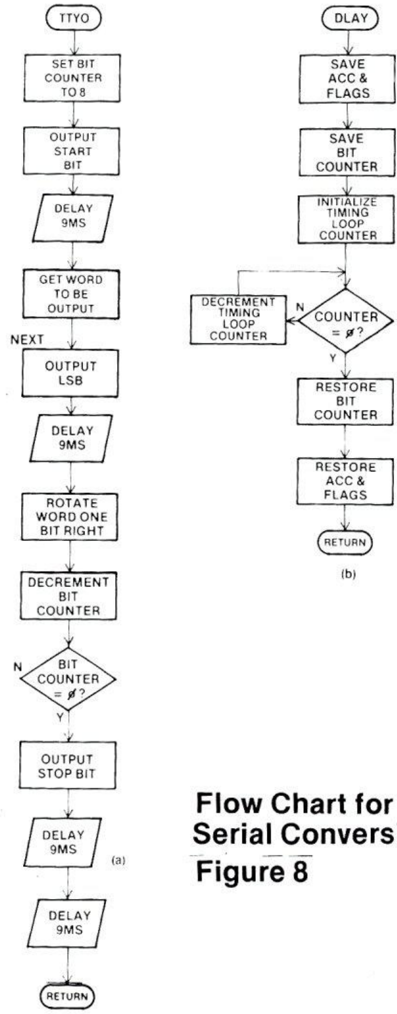
A counter must first be initialized to count the eight bits. Then the start bit, a zero, must be output. (For this example, bit O of the output port will be used as the serial bit and it will be assumed that the other 7 bits are not connected in the system.) A delay routine will then be used to allow 9.09 milliseconds to elapse before the next bit is output. From this point it is an easy task to output a bit, generate a delay, rotate the word for the next bit, and continue until all eight bits have been output. The last steps, lines 13 to 16, set the output bit to the marking state, a one, and delay there for at least two bit periods before another word can be output.

In implementing the program, some consideration should be given to the technique of generating the time delay. One possible approach would be to have a fixed time delay routine of say 9 ms and use the intervening instructions to make up the other .09 ms. Another approach would incorporate a universal time delay routine that would require set-up before entry. For example, in figure 1, the first line of the program would be contained in the main program and the timing routine would be a little more universal since the "COUNT" can be program generated.

In the program example shown in figure 9, I have implemented the "fixed-time" approach, but you will notice a difference between going from line 5 to line 7 (34 cycles) and going from line 8 back to line 7 (46 cycles). The problem is that the time between the 2 delay routines is 12 cycles or 6 microseconds. This is not really serious for one bit, but it means that the last bit will be 8 x 6 microseconds (48 microseconds total) delayed from the intended position. One way to overcome this difficulty is to add 3 "NOP" instructions (at 2 microseconds each) after line 5 to make both paths equal. This same problem occurs again at lines 14 and 15. For these reasons, it might be desirable to use a more universal timing routine and add an "LXI B" instruction just prior to each "CALL" in order to get more accurate timing. This approach, however, adds three instruction bytes per CALL. One other approach would be to use the loop called "NEXT" to output all 11 bits. It's not quite as straight-forward as the example illustrated but it will allow more accurate bit timing for each bit and should reduce the number of program locations. It turns out however, that the slight error introduced in any of the approaches described will not affect the teletype operation. Other timing applications may not be so tolerant.

For more "homework" I will let you generate your own version of a program that can be used to de-serialize incoming data. If you incorporate the techniques discussed in this and the previous article, you should have no problem. (One hint I should give is that it is desirable to read the data bits in the middle of each expected time window in order to allow a variation from bit to bit.)

In the next article, I will incorporate the transmit and receive routines into a single program and will compare these program techniques with hardware implementation. ◘


TTYO
SET BIT COUNTER TO 8
OUTPUT START BIT
DELAY 9MS
GET WORD TO BE OUTPUT
NEXT
OUTPUT LSB
DELAY 9MS
ROTATE WORD ONE BIT RIGHT
DECREMENT BIT COUNTER
N
BIT COUNTER = 0?
Y
OUTPUT STOP BIT
DELAY 9MS
(a)
DELAY 9MS
RETURN
DLAY
SAVE ACC & FLAGS
SAVE BIT COUNTER
INITIALIZE TIMING LOOP COUNTER
DECREMENT TIMING LOOP COUNTER
N
COUNTER = 0?
Y
RESTORE BIT COUNTER
RESTORE ACC & FLAGS
RETURN
(b)

Flow Chart for Serial Conversion

Figure 8

```
Line
No.   Cycles   Label    Operation      Comments
 1     10      TTYO:    LXI H, WORD    ;SET HL TO LOCATION OF BYTE (WORD)
 2      7               MVI B, 8       ;SET BIT COUNTER TO 8
 3      4               XRA A          ;CLEAR ACCUMULATOR
 4     10               OUT 5          ;USE PORT 5, BIT 0, FOR DATA
 5     17               CALL DLAY      ;CALL THE 9.09 MS SUBROUTINE
 6      7               MOV A,M        ;GET WORD TO BE OUTPUT
 7     10      NEXT:    OUT 5          ;OUTPUT ALL BITS (ONLY LSB USED)
 8     17               CALL DLAY      ;WAIT 9.09 MS
 9      4               RCC            ;ROTATE EIGHT BITS
10      5               DCR B          ;DECREMENT BIT COUNTER
11     10               JNZ NEXT       ;IF ALL DONE OUTPUT STOP BITS
12      7               MVI A, 1       ;SET LSB
13     10               OUT 5          ;OUTPUT STOP BIT
14     17               CALL DLAY
15     17               CALL DLAY
16     10               RET            ;TTYOUT IS COMPLETE
17     11      DLAY:    PUSH PSW       ;SAVE DATA IN ACCUMULATOR
18     11               PUSH B         ;SAVE BIT COUNTER
19     10               LXI B, COUNT   ;SET LOOP COUNTER
20      5      LOOP:    MOV A,B
21      4               ORA C
22     10               JZ DONE        ;CHECK FOR COUNTER EQUAL ZERO
23      5               DCX B          ;DECREMENT LOOP COUNTER
24     10               JMP LOOP
25     10      DONE:    POP B          ;RESTORE BIT COUNTER
26     10               POP PSW        ;RESTORE ACCUMULATOR BACK
27     10               RET            ;RETURN TO PROGRAM
```

FIGURE 9

THE COMPUTER MART

One – stop computer supermarket.

IS OPEN

At 625 West Katella Avenue In Orange, CA. 92267

you can call (714) 633-1222

* *

WE NOW HAVE

FOCAL®

For the IMSAI 8080 Computer.

Come in and get your copy of Focal®

On paper tape or cassette $9.00 with instructions.

Get a free source listing of FOCAL$^{\text{T.M.}}$ with any board/kit purchase of $125 or more. This version of FOCAL provided by Processor Technology Co.

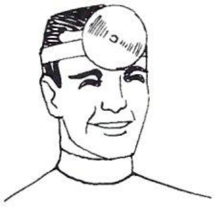
FOCAL Is The Registered TM of Digital Equipment Corporation.

WE CARRY –
- IMSAI 8080 Computer – The Best Way
- Processor Technology Company – The Ultimate
- Solid State Music – Unusual Kits
- Polymorphic Systems – An alternate hard to beat
- Cromenco – The Dazzler® Dazzlers.
- T. I. Sockets the world
- Sanyo Monitors shows the bits and sometimes the byte.
- Vector Electronics for homebrewing

Also complete kit building tools and gadgets.

* *

The Computer Mart The One-Stop Shop

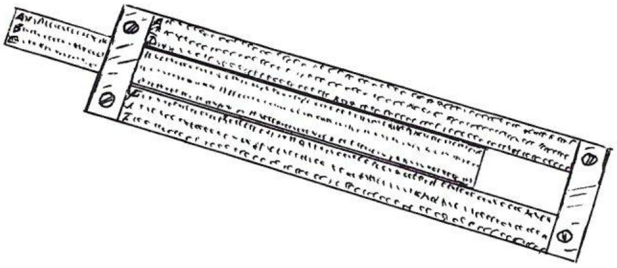


We service what we sell and some we don't sell, we even service _ _ _ _ _ _
George Tate, The Computer Doctor.

CIRCLE NO. 6 ON INQUIRY CARD

Calculatin' Engines Whatever Happened

By William E. Peterson



So what's this all about? This is the first of a series about the history of computing devices. In many ways it's the history of science and tools used in the past to "seek for truth." You will discover the names of eminent mathematicians mingled with obscure ones, instrument makers and descriptions of their art.

The title expresses it. Nowadays "engines" denotes prime movers, yet in the 18th and early 19th centuries, the word stood for all sorts of mechanical contrivances. A device still in use for cutting gears is an *indexing head.* It's used to rotate a metal disc so that each gear tooth occupies an equal angle on the disc perimeter.

In the 18th century the same device was called a "dividing engine". Steam engines retain the original word—indexing heads don't. Calculatin' engines are then "devices for reckonin' and cipherin'" and such.

Our main objective is to *always* keep this series interesting. If there is some device *you* would like to see reviewed, write and we will do an article on the subject. Calculatin' engines have been this writer's hobby and we'll try to pass along some knowledge on the subject in what (we hope) will be a captivating fashion.

WHERE DID THE SLIDE RULE ORIGINATE?

I was saddened to hear that an old line drafting and instrument supplier (K & E) has stopped manufacturing slide rules. They made some of the best, and have been at it since 1886. Everyone knows what a slide rule looks like, but how many know their history?

Who invented the slide rule? Nobody really—it's the outgrowth of the efforts of many men, each contributing to the final product. There is a lot of power contained in that little stick, and it starts with the Baron of Merckiston in Scotland, John Napier. In 1614 he publicly announced his new invention, logarithms. He stated that it was "multiplications, divisions, square and cubical extractions of great numbers" that "doth molest and hinder calculators". He wished to eliminate these things that "are for the most part subject to slippery errors". Logarithms provided the answer "by what certain and ready art I might remove those hinderances." By using logarithms all the multiplication techniques are reduced to addition of logarithmic numbers. Division is multiplication upside down, and so subtraction replaces division in logarithmic computations. We will leave the reader to speculate on the fact that multiplication is merely a short *technique* of addition, and division being then subtraction foreshortened. Mathematics therefore moved forward by stepping back.

Having made this clear we note that Napier played it safe. Ten years before announcing logarithms (in 1594) he privately discussed his invention with the Danish astronomer, Tycho Brahe. Logarithms are the key to the slide rule.

Since the use of logarithms are basically a paper technique, a book of tables is required. In 1620 a bright chap, Edmund Gunter, Professor of Astronomy at Grisham College, London, made a ruler on a strip of wood. It was inscribed with a logarithmic scale. These rulers were known as Gunter's Rule or Guner's Scale. To multiply two numbers, A x B they proceeded as follows. Using a pair of dividers a *distance* equal to the logarithm of number (A) was taken from a Gunter's Rule. This distance was "laid down" on a straight line. Then another distance equal to the logarithm of a second number (B) was "laid down" end to end with the first. By measuring the total distance with the Gunter Rule, voila, the product of A x B appeared. Trouble was it was messy, dividers and rulers and all that, though faster than puttering through books of tables. The first step to cut "computer time" had been taken.

Living close to London was a very clever cookie, Rev. William Oughtred. He took two Gunter Scales and placed them edge to edge. By sliding them against one another lengths A and B could be added *without* dividers. His friend William Forster wrote about this innovation in 1632. Rev. Oughtred also did something no one else had thought of, making two circular disks pivoted at the center with logarithmic scales on their perimeter. The circular slide rule was born. In addition to the logarithmic scales, he added logarithmic trig scales. The race was on!

In 1657 Seth Parthridge, a mathematics teacher and surveyor, made what would be called today the common slide rule. Three strips of wood laid side by side, the center one being the slide and the two outer strips bridged by small sections at the ends. This rule contained scales on *both* sides. Technology was getting complex.

To The Slide Rule?

In 1683 Thomas Everard (an excise officer at Southhampton, England) brought out his "sliding—rule". He coined the name of this new device. Government decrees not withstanding, words with many syllables usually get shortened through popular usage. "Slide-rule" (later without the hyphen) emerged as the common name. Commercially oriented Everard added "gauging points" or marks to the scales. These marks indicated often used settings for conversion factors then in commercial use. Also included were scales for square and cube roots. This type of rule was therefore a commercial or "gaugers" rule, rather than a specifically scientific device. Technology had reached the market place!

Another fellow now appears on the scene by the name of William Hunt. The year is 1697 when he starts tacking scales on, right and left; a scale for the area of circles if the diameter is known, a scale for finding the perimeter of an ellipse if the axes are known, and a scale that's still with us today, the reciprocal scale.

With all this bustle of activity taking place in England, let's see what's happening on the continent. The idea of the Gunter's Scale appeared in France in 1624 due to an English writer, Edmund Wingate. As a matter of fact, he was so prolific (in writing) that for many years he was considered the inventor of the slide rule.His main contribution was an added scale. Wingate's scale was an equally divided one (or linear scale) and therefore logarithms of numbers could be seen at a glance.

Slide rules were not known much in Europe before 1700. The first German writer on the subject, Biler, wrote a book in Latin on mathematical instruments in 1696. For some reason, slide rules were not popular in Europe even in the 18th century, but in insularly England they thrived.

By the year 1700, in England, the slide rule was carving its place. This was mainly due to better education, since the use of fractions was used and taught there to a greater extent.

While we have talked about "sliding rules", mention must still be made of the hair line runner. Sir Isaac-Newton, while working on the theory of algebraic equations devised a special rule for solving cubics. This had three movable scales. A hairline runner was placed across the entire rule for ease of reading three numbers simultaneously on the three sliding scales.

This concept was first published by E. Stone in 1726. By 1775, John Robertson, mathematics teacher at the Portsmouth Royal Academy was attaching this auxiliary hairline to the frame as an integral part of the rule. Also, in this year James Watt (of steam engine fame) has special ordered precision graduated rules. This raised the overall quality of the product and promoted its use by English engineers.

This brings us to the year 1800. Recollect that England was predominant in the construction, promotion and development of calculatin' engines prior to 1800. In the 19th century English predominance declined. The last of the great innovators of slide rules in England was William Nicholson. He published a technical journal called "Nicholson's Journal", which contained advertisements for various slide rules. In addition, Nicholson tried for further developments and encouraged the use of the devices. Circular rules and spiral types became his specialty. Though he died in 1815, his journal carried on and in 1817 one Sylvanus Bevan introduced the folded scale. This scale was basically a logarithmic type scale broken in the middle and the two pieces stuck end to end. Now with careful layout on a logarithmic scale the center of the scale is pi (3.1416, etc.). Hence the folded scale is folded at the value pi. By doing this, calculations involving pi are quickly solved, since any number multiplied or divided by pi can be quickly set on the normal and folded scales. Hence the slide need not be used. This speeded computation.

DECLINE AND REVIVAL

Slide rule development and use declined in England after this period. However, with the rise of Napoleon in France the innovative slide rule was also on the rise. With the collapse of Louis XVI in France (and the French revolution) old ideas were abondoned. With this turning to new ways (the metric system, for example) the use of the slide rules found fertile ground.

Peter M. Roget, a French physician invented the log-log scales. In 1815 he reported his invention of this "logometric" scale to the Royal Society of London. Though the log-log scale can display *any* base to *any* power, there was not much need of that in those days and this scale fell into neglect and was soon forgotten.

By 1900 computations requiring this use of any base to any power were appearing throughout physics, and as a consequence, Roget's log-log scales were re-invented.

SOPHISTICATION

A practical invention in 1821 by Lenouir of France was a machine for graduating the slide rule scales. Up to 8 slide rules could be simultaneously marked off. Mass production therefore began. The accuracy was also tremendously improved with expert machine graduating.

In 1843 Austrian Professor, Von Strasnitzki, of the Royal Institute at Vienna, made a large slide rule 8 feet long for classroom demonstration. Classrooms the world over have such large working models to this day.

ANALOG COMPUTER

And now we come to the greatest step taken in the development of the slide rule. Prior to 1850 the slide rule scales were used to solve specific short calculations. In 1850 a Lieutenant of the French Artillery, Amedee Mannheim by name, conceived a new view. By layout of the scales already in use, in a specific manner the slide rule became an *analog computer*. The layout of the slide rule scales according to his pattern of 1850 is still used, though few realize why. This later led to Mannheim becoming a professor of mathematics at L'E' cole Polytechnique in Paris. In 1850 when 19 years old and an artillery officer at Metz, Mannheim completed the layout and published a book of instruction. Though the Mannheim rule was adopted as standard in the French artillery, it was not until 1890 that it appeared to any extent in the United States.

HOW IT WORKS

Let's take a look at how to use the slide rule as analog computers. The Mannheim layout technique of the various scales is still in use, so if you will pick up your slide rule we'll start.

Remember that Mannheim was an artillery officer. In 1850 trajectories of the cannon shot in use were computed by just such officers. These trajectories represent a parabola, a specific second degree equation. One characteristic of the Mannheim layout was such that all real roots of second degree equations could be rapidly found. Second degree equations are conic sections and include the parabola, hyperbola, circle and ellipse. The slide rule can rapidly find real roots (imaginary roots aren't too useful in an artillery section when somebody else is shooting back). Getting the range from the trajectory represents a special case of "get there the quickest with the mostest", hence the French artillery adopted "the quickest" computer, i.e., Mannheim's slide rule.

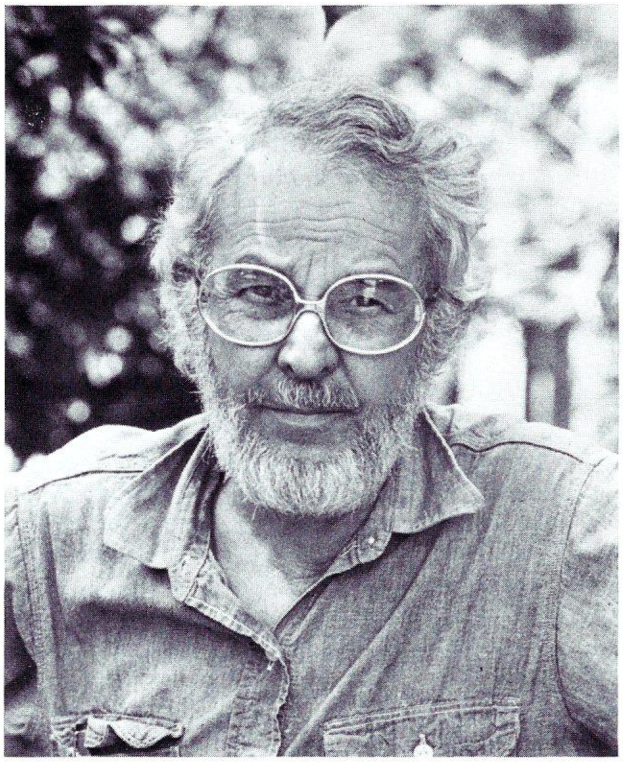
How to do it: take a look at your slide rule. On the frame is the D scale and on the slide is the C scale. Just above the C scale is the CI (or inverse) scale. Both C and CI are on the slide. Lets's call the right hand index of the C scale C10 and the left hand index C1. To rapidly multiply two numbers, say 2x4.5, put the hairline over the 2 on the D scale. Move the slide until 4.5 appears under the hairline on the CI scale, read under C10 on the D scale the answer 9. Now, if our slide rule had no square root scale we could still find the square root of 9 by proceeding as follows. Leaving C10 over 9 on the D scale, move the hairline over C10. Now, slowly move the hairline to the left while watching the numbers on the CI scale and D scale slowly pass by. Notice that the CI scale numbers get larger while the D scale numbers get smaller. Eventually you will reach a point where they are equal (at 3). This is the square root of 9. Notice that this technique of watching the CI and D scales is an analog process that led us to the solution of the specialized 2nd order equation $Y^2 = 9$. Try finding the square root of 5 by this technique (the root is 2.24). You will find that when comparing the CI and D scale numbers you must first keep track of the most significant digit (2), then the next most significant (2.2) and last, the final digit of resolution (2.24). Now what if we had a problem like $Y^2 + 3Y = 5$? This is really $(Y)\ (Y + 3) = 5$. For the solution set C1D over 5. Move the hairline, watching the CI and D scales. Your objective is to have the CI reading be $(Y + 3)$ or 3 units larger than D. This is where it pays to watch the significant digits. The final answer arrives when CI has a reading of 4.19 and D is 1.19.

To determine the sign (plus or minus) of these roots, factor the initial equation $Y^2 + 3Y - 5 = 0$. The sum of these factors must equal $+3Y$. Hence the sign of the factors must be $(Y + 4.19)\ (Y - 1.19) = 0$ and the roots are $Y = -4.19$ and $Y = +1.19$. Here we have solved a quadratic equation with the CI, C, and D scales. Mannheim had the A (or square root scale) on the mainframe. Using the A scale and the CI and D scales it is possible to solve cubic equations, though this operation is more complex.

DEMISE

And so let's doff our caps to the slide rules' demise; mystical land, like the graveyard of last ships, or the dying elephants in a valley of ivory. As I have tried to illustrate, the slide rule as an analog computer has a place that can't readily be filled by a cheap, non-programmable calculator. With no batteries, no buttons to jam, no chips to pop, there ain't much profit in 'em. Some day 2,000 years from now, an archeologist will exhume a slide rule and exclaim, "Look, a religous artiface! Ah, what primitive people they were." ◘

THE RAW AND THE COOKED SIT AND DANCE OR DANCE AND SIT



A portrait of the author.

by John Whitney

The following paragraphs are taken from a talk which I was asked to give at a conference of industrial designers in 1962. I chanced to look over that paper the other day as I was beginning to reflect upon the best way to start a discussion of the formal design problems that will probably underlie our study of computer graphics. Here was an attempt from fourteen years ago to outline some basic considerations for "motion graphics".

Motion graphics was my term then and I'm afraid it is no better than a dozen other terms that have been invented since. I doubt if we have yet found the name which will permanently identify the raw world of art which we are beginning to explore with computers and graphical output devices.

As a beginning, and to provide some background on all this, and to demonstrate a measure of the continuity of the ideas, I present these paragraphs from the past.

When a major feature film title in the tradition that Saul Bass has done so much to establish, has interesting articulation, it usually succeeds as a title. On the other hand, if it is lacking effective motion or articulation it might as well have been a book jacket. This matter of articulation is what I refer to as *Motion Graphics* and it is distinctly a new problem in the field of design; so little explored in fact that designers must approach it with caution and the proper sense of adventure.

Now how does one delineate motion graphics? Abstract film-makers have been pondering this question; stabbing at it in the dark of projection rooms since the early twenties. Man Ray achieved a brief triumph in this direction forty years ago. He simply stood starched white collars on end on a phonograph turntable (a trick in itself) and filmed the turning spiraling white shapes against a black background.

About every abstract film-maker since, who has had his own modest success, has come by it with almost as much fortuitousness. That is putting it rather crudely; but it is so that some have achieved high moments on film by swirling a mixture of incompatible paints in a barrel, or scratching, biting, crayoning and otherwise torturing raw motion picture film without using a camera at all. Others have hung other people's sculpture on invisible strings and let them turn in front of a camera, still others have filled space with wads of colored plastic and such and set spotlights moving and filmed the reflections and refractions moving on the wall.

I have listed this hodge-podge of techniques to emphasize that if you want to compose an abstract film you just can't pick up any thread of tradition winding back into the past. *And you can't walk into a store and buy an instrument that plays abstract design.* A camera is of no use in this respect; more than a phonograph would be of use to a composer. Abstract film making then (the pure art of motion graphics as opposed to film titles, which is an application of the art) is a *new* art which demands *complex tools.*

Now to digress a moment: Before the war, I contracted what at that time was a common fever to make film "Symphonies," Joris Ivens and others made these films . . . and always called them "Symphonies," (just why-I couldn't say since they were always such little films-hardly a movement of a Symphony). I found that if you go out with a motion picture camera and shoot nature, you'll be surprised how little real *action* you can get. Especially if you know as anyone should that Slavko Vorkapich had already shot all the ocean and clouds that need shooting for all time-ever! And Ivens had already shot too many rain drops and riverlets. "What else on earth moves?" you ask yourself. Nature, all in all presents a surprisingly static image to the cameraman who wants to make "Symphonies." Abstract ones, that is, where animals and humans don't fit.

I have brought this up only to make the following ironic observations that our *concepts* of nature and the universe have become dynamic. We know in this Non-Aristotelian age (to borrow a phrase) that time and motion or, time/motion characterize modern world.

Now I might quote here from Gyorgy Kepes' *The New Landscape* or books of art philosophy. Instead, I need only go to the grocery store and pick up a copy of Harper's and read to you what Sir Kenneth Clark of the London National Gallery has to say. I quote:"Art and science are not, as used to be supposed, two contrary activities, but draw on many of the same capacities of the human mind." Sir Kenneth writes, "Artists and scientists alike are trying to give concrete form to dimly apprehended ideas . . . science is the search for unity in hidden likeness, and the starting point is an *image* because then the unity is before our mind's eye." He gives us the example of Copernicus' notion of the solar system which was inspired by the old astrological image of man with the signs of the Zodiac distributed about his body. Then Sir Kenneth continues: "Our Scientists are no longer as anthropomorphic as that; but they still depend on *humanly comprehensible images,* and the valid symbols of our time, invented to embody some scientific truth, have taken root in the popular imagination."

Here in a nutshell is background for abstract art. But notice that here we can also perceive the timely validity of the abstract image in motion. The nuclear scientist will agree readily that he is concerned with dimly apprehended ideas of, *forces and energies and particles in motion.* His artist contemporary should (and does not) have any great facility to create image in *motion.* To repeat: Our concepts of the universe have become overwhelmingly dynamic-the valid symbols of our time possess this dynamism implicit in them-I am concerned with *motion graphics* because I feel the need for explicit *motion* in *graphics* as against the implicit motion of the painter's canvas.

The search for a viable system of motion graphics has ranged far and wide among abstract film-makers. Here and there we have had it on our conscience that we had become too much involved in mere technology. It has been a sensitive point with some to hear a challenge, that goes like this: "If you are so damned interested in this kind of film why don't you make films, instead of messing around with all the gadgetry." For better or worse, I have resolved this question in my own mind. I decided last year to engage in the construction of bigger and more eleaborate gadgets.

My points of departure would seem to be at the crossroads where computer designers abandoned mechanics altogether and introduced electronics. This turn in itself was fortuitous, since it made obsolete a $33,000 device which was then placed in surplus which I can now buy for less than $200. I am told it is one of the more sophisticated mechanical analogue computers of specialized function that were produced during the last war. It happens to be an "Anti-Aircraft Gun Director." It contains a beautiful network of cams that are variable and make an exceedingly versatile two dimensional coordinate design instrument—with a few alterations. I have not studied the history of cam linkage deeply, yet it is interesting to observe again and again all the way back through the 18th Century that cams have been assembled variously to trace intriguing designs on paper. Yet through all this time no one seems to have found much use for these designs other than to decorate gilt edge bonds and currency.

At any rate, I am trying to erect a total motion graphic system. If I were a composer like Arnold Schoenberg, at least I might arrive at a system without having to invent the symphony orchestra in order to hear my system applied. Yet that advantage already has been lost to many modern composers. Karlheinz Stockhausen must penetrate deeply into the field of technical invention if he persists with electronic music. So I am not alone when I find that if I *must* make a system, I must also simultaneously invent machines.

Now let me try to explain some of the philosophy of a "gadgeteer" and I will be through:

In the first place we need only look at the twelve tone scale or the alphabet to realize what versatility is possible from a permutational system. So I am looking for a permutational system of graphics. The first of the little silent films shown here tonight is an example of rudimentary, permutational graphics. The entire film was made from one rectangle and one circle in one position in the film frame. All other positions of the circle and rectangle are achieved by inversion and mirroring the film strip from the original positions. The action too is a serial configuration that undergoes permutation by reverse direction, skip printing etc.

But this film suffers from limitations which suggests that as a system of composing it could soon be exhausted of fresh possibilities and thus die. So the second requirement for a system of graphics might state that it must be viable. Somehow the ideal system must offer resistance so that with every new approach the composer finds before him a challenge.

For example, risking over-simplification of history, it may be said that after a few hundred years during which western musical culture evolved this challenge in music composition has come to resemble nothing so much as a tidy problem in mathematics. Rules have accumulated in profusion and the challenge is there as any first year harmony student will agree.

Thirdly, I am looking for a graphic system that will produce simplicity and complexity at one and the same time. This would be hard to explain in terms of graphics. Another meaning of the word "articulate" refers to speech; to enunciate distinctly. Speech poorly articulated is garbled speech. The communication, specialists refer to all elements in a message that do not contribute to the message as noise. Now for the purely sensuous appeal of texture and pattern there is need for complexity in motion graphics. But simultaneously there is need for simplicity for the sake of articulation and communication. Still there must be no noise in the system.

Now what constitutes noise if the message happens to be a music composition? Of course, it might be a jet passing overhead or hum in the amplifier. But it could be clumsy orchestration in which the clarity of the musical ideas have been simply spoiled with unnecessary doubling of voices and generally inept composing. This too is noise, I think, by the communications definition.

Inept and unnecessary embellishment also spoil the message in graphic art, obviously. Yet judicious use of texture and pattern can be an aid to the message in graphic art, obviously. Yet judicious use of texture and pattern can be an aid to the message. In motion graphic art, obviously. In motion graphics the problem is compounded by the fact that all texture and pattern can be an aid to the message. In motion graphics the problem is compounded by the fact that all texture if it is static, looks static. There is a disparity between action and any static element in the frame that always seems irreconcilable and therefore contributes only noise. The problem of dynamic texture that the animator is familiar with is only magnified here. Thus for the sake of unity and clarity of the message, action in terms of the total frame must be considered. So, the third feature of my system seeks to involve the whole frame in a species of multiplex action that still must be simple and unified in its total effect, looking for a graphic system . . . both simple and complex.

My ideas in 1962 were influenced by the writings of the painter Piet Mondrian, his concepts of reconciled force, his "equilibrium" which was to be achieved by the dynamic balance of opposed forces.

The painter is a juggler who balances colors, shapes and implies forces of direction and tension. The fruit of his juggling act is a static fixed monument. The painting is a frozen record of his act. For example, with the abstract Expressionists—the "action" painter—the canvas had become a metaphore of a gesture; a mute record of actions and tensions that were expressed once; a dynamic action event that is now over and past.

With us computer freaks it has to be the other way around. It doesn't take physical gymnastics—slopping paint on canvas and so on—to program a computer. Nor do I think we want to end up with a static, frozen image on canvas. Our interest and our requirements are decidedly changed from the painters. They jump around a lot and produce something that just sits there. We may sit quietly and produce some pretty fancy dancing on our computer screens.

We must learn to show a balance between tensions and forces of shape and direction and color . . . all of this in a context of metered time. Metered time is orderly time metered out "byte for byte" like words, sentences and paragraphs. That is to say we must create the image of time which is given the proper shape for human consumption.

And we have to learn what the composer already knows—how to reconcile the human need for diversity with the other needs for familiarity. Composer, prostitute and shoe salesman all have it in common—create a demand and satisfy the demand.

All these tricky little arts we must learn to achieve with our present knowledge and our current primitive, non real-time computer equipment. This is a staggering challenge. But the very fact that we can visualize these possibilities today suggests that we are committed to meet that challenge. I hope to be able to present some clues as to how some of those problems may be dealt with next month. ■

SCCS PRODUCES OWN SLIDE SHOW

A slide show is being produced for the SCCS hospitality suite in March at the Altair Convention. You are probably knee-deep in your own computer project . . . we can see it all now . . . you, amidst circuit boards, wires hanging from everywhere, transistors and chips askew. Why not take a 35mm color slide or two of your baby, write up a brief explanation of what you're into, and send it to coordinator Ruben Loshak. He'll greatly appreciate your involvement, and viewers of the slide show will benefit with a better understanding of what SCCS and home computing are about. The show, which will be produced under the talented direction of graphics designer, Tom Woodard, will feature short taped interviews to accompany the slides presented. Photographers and interviewers with tape recorders who wish to help are invited to contact Ruben. Write him at 4040 Badillo Circle, Apt. #50, Baldwin Park, Calif. 90706, or call (213) 338-8549.

New Products

NEW "INTRA-SWITCH" CAN OPEN/CLOSE INDIVIDUAL RIBBON CABLE LINES FOR TESTING

A unique, long-needed device for instant, open or closed switching of **individual** lines in flat ribbon cable systems is now available from A P Products Incorporated. Diagnostic testing of circuits on a **line-by-line** basis is now possible using the A P "INTRA-SWITCH".

Applications include switching commands to various control boards as well as switching I/O signals to or from test equipment. Other uses include programming preset logic functions into systems.

The A P Intra-Switch installs manually in seconds. It mates with standard dual-row connectors and sockets with contacts on .100" matrix.

The recessed individual slide switches are easily actuated with any pointed device such as a pencil or probe tip.

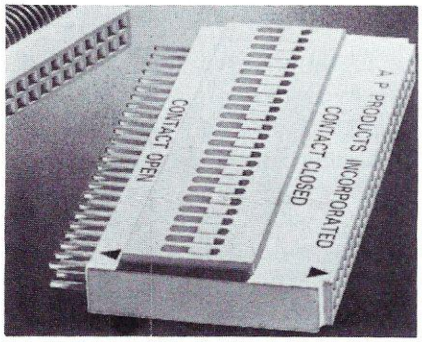
Five models are available for the most common flat cable connector/socket sizes: 20, 26, 34, 40 and 50 contacts or lines.

Contacts are rugged, non-corrosive, alloy 770. Dielectric body is glass-filled polyester.

Prices start at $12 for the 20-line model.

Availability: Stock to four weeks. Available through all A P Distributors.

For further information, contact R. Gabor, A P Products Incorporated, Box 110, 72 Corwin Drive, Painesville, Ohio 44077, phone: (216) 354-2101, twx: 810-425-2250. Direct all inquiries to Rita Mercer.



CIRCLE NO. 91 ON INQUIRY CARD

EBKA "FAMILIARIZOR" MICROCOMPUTER

EBKA introduces the "Familiarizor", a complete microcomputer system with its own hexidecimal keyboard and display built into a single printed circuit board. The total package includes self-teaching hardware and programming manuals containing step-by-step instructions which provide a practical, low-cost introduction to microprocessing combined with "hands-on" experience.

The heart of the EBKA"Familiarizer" is an MOS Technology 6502 Microprocessor, an eight bit processor that can address up to 65K bytes of memory. On-board memory consists of 1K bytes of RAM for user programs and two eight bit input and output ports. A 256 byte monitor program, supplied in one 1702A erasable PROM, and an on-board terminal replaces more complex lights and switches to permit simple loading, examination, running, debugging and modification of programs.

To store programs in PROM, the PC board is designed to accept three additional 1702A PROMS (768 bytes). A low cost PROM programmer is also available, as well as add-on memory, interface and special function cards to expand the capabilities of this new microcomputer, in easy and inexpensive stages.

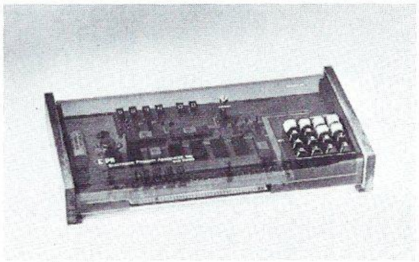
The "Familiarizor" is available in kit form including all parts, manuals and documentation for $229. Completely assembled, the cost is $285. Optional power supply is available at $85. Write EBKA Industries, Inc., 6920 Melrose Lane, Oklahoma City, Oklahoma 73127, or phone (405) 787-3671 for more information.

SOFTWARE MANUAL
HARDWARE MANUAL
EBKA

CIRCLE NO. 92 ON INQUIRY CARD

COMPLETE MICROPROCESSOR AVAILABLE

Electronic Product Associates, Inc. has introduced their MICRO 68 computer system priced at $430, assembled. Built around the Motorola/AMI 6800, this system includes an integral power supply, 16 button keyboard, 6 digit LED display, 128 words of RAM, PROM, 64K memory expansion capability, and connectors for 16 bit I/O. It's packaged in a 9x16x2 hardwood cabinet with transparent smoke plexiglass lid. Obtain more information by calling Vic Wintriss (714) 276-8911, or write Electronic Product Associates at 1157 Vega St., San Diego, Calif. 92110.



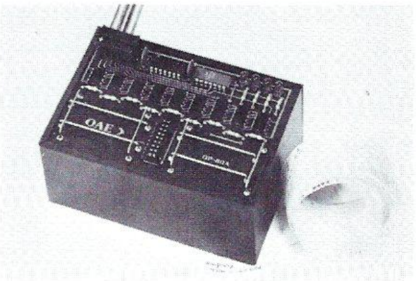
CIRCLE NO. 93 ON INQUIRY CARD

OP-80A HIGH SPEED/LOW COST PAPER TAPE READER

Pictured above is the new OP-80A High Speed Paper Tape Reader from OAE. This unit has no moving parts, will read punched tape as fast as you can pull it through (0-5,000 c.p.s.), and costs only $74.50 KIT, $95.00 ASSEMBLED & TESTED. The unit is small, light, and portable. It includes a precision optical sensor array, high speed data buffers, and all required handshake logic. The unit will interface directly with an 8 bit uP I/O port, or can be connected across a UART allowing you to load programs through the TTY I/O port directly without software modifications!

The OP-80A is ideal for loading all types of microprocessor support software, including monitors, assemblers, editors, memory and hardware test routines, field test rountines, games, etc. And FAST! (You can easily load 12K Extended BASIC in 30 seconds!)

The kit includes four status LEDS, an anodized aluminum box, four foot flat interface cable, assembly instructions, interface schematics, and a bootstrap loader program. Available from: Oliver Audio Engineering, 1143 N. Poinsettia Dr., Los Angeles, Calif. 90046, (213) 874-6463.



CIRCLE NO. 95 ON INQUIRY CARD

TWO-SIDED DIP BREADBOARD PROVIDES MORE CONNECTION POINTS

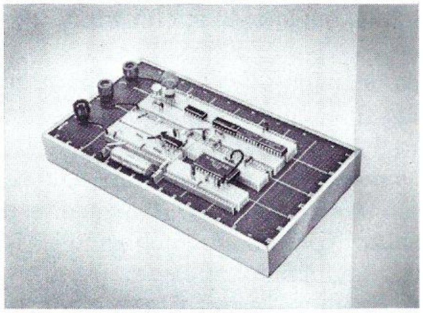
A versatile new Klip-Blok breadboard, mounted on an aluminum chassis allows components to be attached and interconnections to be made from both sides to increase wiring convenience and reduce clutter. Signal and power bus connections can be made on the bottom, if desired, leaving the top clear for wiring that must be altered. Interconnections can be made with 22 gauge wire. DIPs and most semi-conductor units plug in.

Designated the Model 51X by Vector Electronic Company, the board assembly is based on the Klip-Blok concept that allows quick, solderless assembly of breadboard circuits. The eight Klip-Bloks accommodate a maximum of twelve 14- or 16-pin DIPs, or four 24- or 40-pin devices. Klip-Bloks may be quickly repositioned for 0.6 in. spacing of tabs. Additional Klip-Bloks, sockets or discrete components may be mounted on the perforated Vectorboard® surrounding the Klip-Blok area. The 51X accomodates packages with 0.6 in. lead spacing as well as devices with conventional 0.3 in. spaced leads. Klip-Strips supply convenient signal or power busing.

The glass-epoxy mounting board is 4.5 in. by 8 in., and has a 0.1 in. spaced hole pattern. Printed registration numbers along the edges identify component positions. The 51X board is unclad; however, a model 51X-GP may be ordered with an etched ground plane on the bottom side for breadboarding high-frequency circuits.

Klip-Blok breadboards are immediately available from Vector Electonic Company, or through the firm's distributors. The 51X is priced at $25.50; the 51X-GP is priced at $29.95. Other sizes are available, as well as individual Klip-Bloks and -Strips.

Vector Electronic Company, Inc., 12460 Gladstone Avenue, Sylmar, California, 91342, (213) 365-9661; TWX (910) 496-1539.



CIRCLE NO. 98 ON INQUIRY CARD

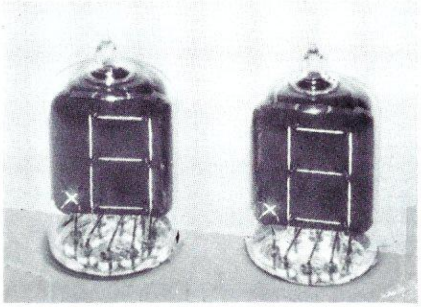
NUMITRON REPLACEMENTS

Industrial Electronic Engineers, Inc. (IEE), supplier of digital readouts and information display devices and systems, announces the addition to its IEE-APOLLO line of Models DA-2000 and DA-2010 Incandescent Digital Displays, direct replacements for the RCA NUMITRON DR2000 and DR2010. The new tube displays are mechanically, electrically, and visually interchangeable with their RCA counterparts, providing OEM's and users with a viable, alternative source for these high brightness devices.

Operating typically at 24mA per segment current and from 3.5 to 5.0 VDC these low-voltage devices have a .600" high, 7-segment character and produce 7,000fL of luminous intensity with a 30:1 contrast ratio. Brightness is adjustable and permits the use of color filters. The DA-2010 features a left-hand decimal point. Both tubes utilize a 9-pin circular base design for installation on PC boards with .8' C/C spacing and many commercially used sockets.

Life expectancy of the design is 100,000 hours and permits wide angle viewing through 140 + °. Fully compatible with IC driver/decoders, the DA-2000 can be either directly driven or multiplexed. Display rise and decay times are 15 and 20ms respectively.

Both devices are available through distributors and from the factory at $2.65 and $2.80 respectively in 1,000 quantities. Contact Robert Brandt, Product Sales Manager at IEE, 7740 Lemona Ave., Van Nuys, Calif. 91405, or call (213) 787-0311.



CIRCLE NO. 94 ON INQUIRY CARD

CIRCLE NO. 96 ON INQUIRY CARD

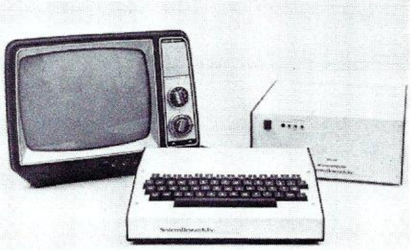
SYSTEMS RESEARCH 8 BIT MICROCOMPUTER SYSTEM

New! From Systems Research Inc. 8 Bit Microcomputer System.

The SRI-500 Microprocessor board, the heart of the system, contains a unique 3-"chip" scheme. One is the CPU (Central Processing Unit), another is the MI (Memory Interface), and probably the most important is the PSU (Program Storage Unit), which incorporates a special Debug program called FAIRBUG.

FAIRBUG provides the user with a convenient and powerful programming DEBUG facility which is used to aid in the development of programs.

Write for details to Systems Research, Inc. P.O. Box 151280, Salt Lake City, Utah 84115, (801) 942-1093.



CIRCLE NO. 97 ON INQUIRY CARD

Cancer's warning signals:

1.
Change in bowel or bladder habits.

2.
A sore that does not heal.

3.
Unusual bleeding or discharge.

4.
Thickening or lump in breast or elsewhere.

5.
Indigestion or difficulty in swallowing.

6.
Obvious change in wart or mole.

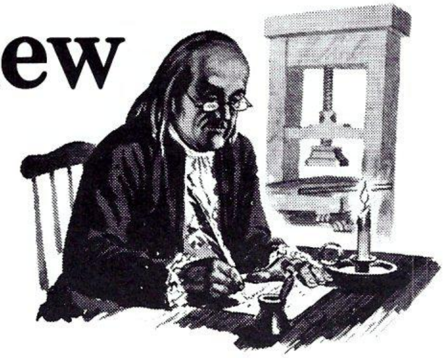
7.
Nagging cough or hoarseness.

If you have a warning sign, see your doctor!

AMERICAN CANCER SOCIETY

This space contributed by the publisher.

Book Review



"MICROCOMPUTER DICTIONARY AND GUIDE"

by Charles J. Sippl
Matrix Publishers, Inc.,
Champagne, Ill. 61820
Price: $17.95 suggested retail

An unbelieveable compilation of 5000 terms, definitions and product procedure and ap
tions. This book is virtually indispe
become an integral part of every no
perienced and professional in the
domain.

It covers all you ever wanted to k
computers, but were afraid to ask ar
book is divided into eight major secti
Micro-electronic terms; Symbols, Ur
of Electronics; Statistics Definition
Computer Acronyms and Abbrevia
Language Summaries, APL, Basic a
puter Number and Binary Switching
finitions of Programmable Calculator

This text is an excellent reference
lute *must* for the serious minded co
rently available in the west from I.C. I

"AN INTRODUCTI
MICROCOMPUTE

by Adam Osborne & As
2950 7th Street
Berkeley, CA. 94?
Price: $7.50

Reviewed by
Carl David Todd, P.E.

The book attempts to give the reader a thorough understanding regarding microcomputers and includes some fundamental information for the reader with no computer background. The step between no knowledge about computers and a thorough understanding is a big one, and I would have to express some reservation as to whether the book will accomplish the stated goal with the average reader.

The physical size of the book is somewhat deceiving in that the subject matter covered is complex and the treatment is in condensed form. I fear that many times the reader will feel short-changed in the brevity of material on a given subject, but a more detailed dissertation might well fill many volumes, each much larger than the present book.

The book is printed in two type fonts with bold face type being used for a coarser treatment with light type employed for supplemental discussion. Unfortunately, the light type is so thin that it becomes somewhat difficult to read on the glossy paper unless the lighting is just right.

The first chapter attempts to define the term "microcomputer" after stating "a minicomputer is a minicomputer because the product's manufacturer calls it a minicomputer." The author leans toward the microcomputer as one or few logical devices.. By his thinking the LSI-11 is not a microcomputer (even though its manufacturer calls it one!). It appears than in many instances, the author uses the term microcomputer where others prefer to use "microprocessor" to describe the single-package arrangement of logical devices.

sents fundamental concepts which
ıl to the reader unfamiliar with basic
ı good review or reference material
may have forgotten some basic infor-
rmation is well organized and lays a
The reader might be tempted to skim
:l lightly, but should not unless he un-
ɔughly.

ıges of chapter 2 contain some errors
ed to thoroughly confuse the reader.
y true where the very important De
m is presented. Any book of this type
to contain some typographical errors
ite a few sprinkled throughout, usually
huckle but little confusion. The prob-
ever, seem to be more than inconse-
ıphical errors. Perhaps "Interface"
lish a corrected section on De Morgan's

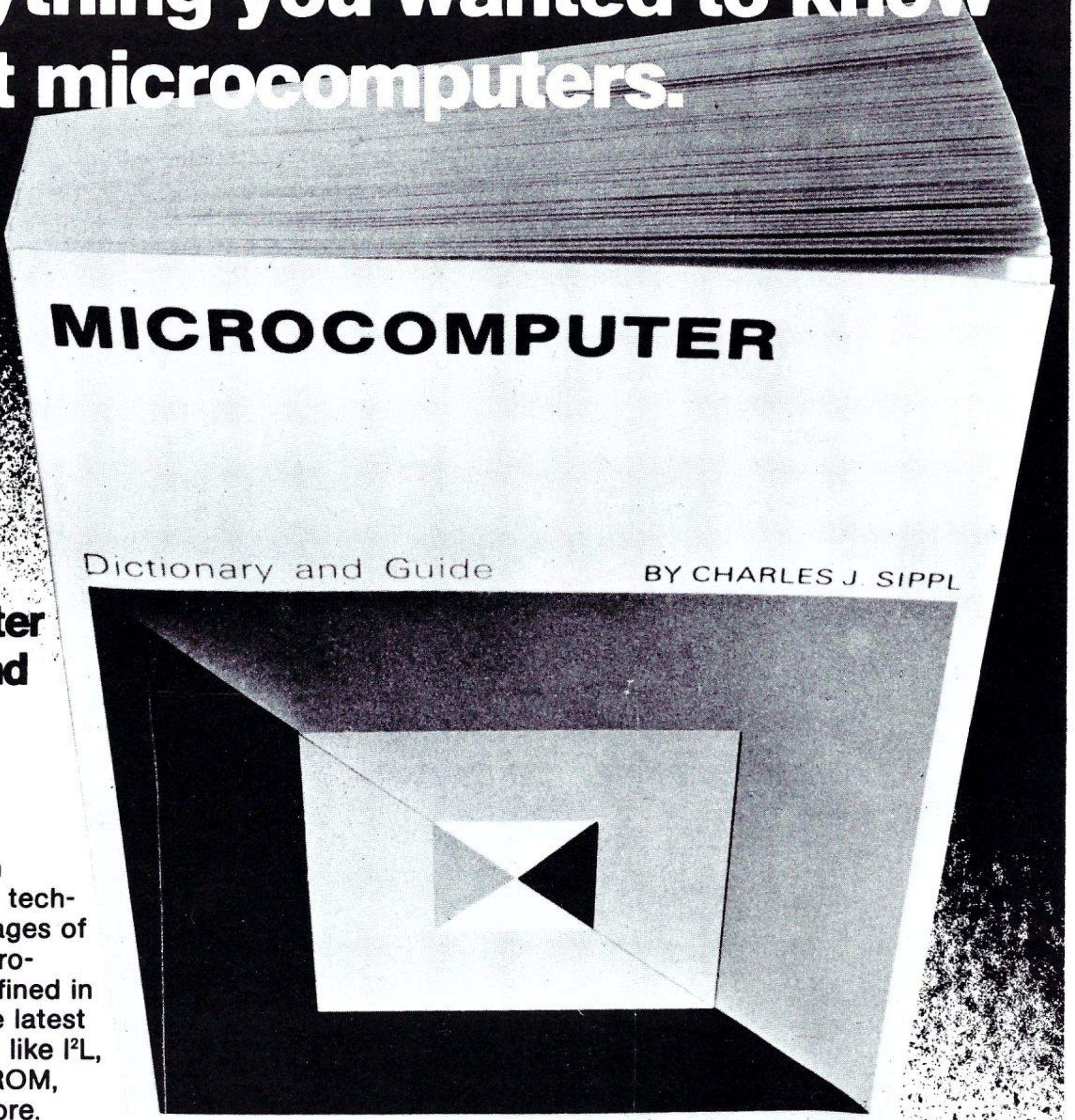
:usses memory types and organization
ı giving understanding of just what is
emory land."

ıkes some incorrect generalities re-
economics, but it's sometime difficult to predict just what will be the lowest cost path between two possible approaches. He also makes the statement concerning static rams that "once data has been written into it, the data will stay there forever." (!!!) Perhaps he has never tripped over his line cord after loading in basic!

In the section on multibyte binary subtraction, the author leads the reader somewhat astray by stating that only the low order byte needs to be twos comple-

(Continued on page 58)

Everything you wanted to know about microcomputers.



Microcomputer Dictionary and Guide

By Charles J. Sippl and David A. Kidd

Come to terms with today's hottest new technology. Over 500 pages of microcomputer/microprocessor terms defined in plain English. All the latest buzzwords included like I²L, CCD, DFT, PLA, EPROM, plus over 10,000 more.

In addition, there are over 200 pages on related subjects, including: mathematical definitions, a study of acronyms and abbreviations used in electronics, a treatise on computer languages and number systems, plus a 38-page appendix on programmable calculators.

This new **Microcomputer Dictionary and Guide** is a must for anyone desiring to stay with the times and join this latest computer revolution.

REGULAR PRICE $17.95

WITH THIS AD

$14.95

I.C. ELECTRONIC SUPPLY

OPEN 7 DAYS

BankAmericard - UPS - Master Charge

Hours 9 to 9 Mon.-Fri., 9 to 6 Sat. & Sun.

Van Nuys, California 91406 16723 Roscoe Blvd., **(213) 894-8171**
Orange, California 92667 619 W. Katella Ave., **(714) 997-5343**
San Diego, California 92111 4861 Convoy St., **(714) 292-5484**

Call or Mail to any of our 3 locations

I can't make it into one of your stores so please RUSH me a copy of the **Microcomputer Dictionary.** If payment accompanies order I.C. will pay for shipping. (That's nice.) California customers add 6% sales tax.

MAIL TO ANY OF OUR 3 LOCATIONS

NAME______________________________

ADDRESS____________________________

CITY________________ STATE________ ZIP________

☐ B of A ☐ Mastercharge # ____________________

Good thru____-____ Please send me______ **Microcomputer Dictionaries**

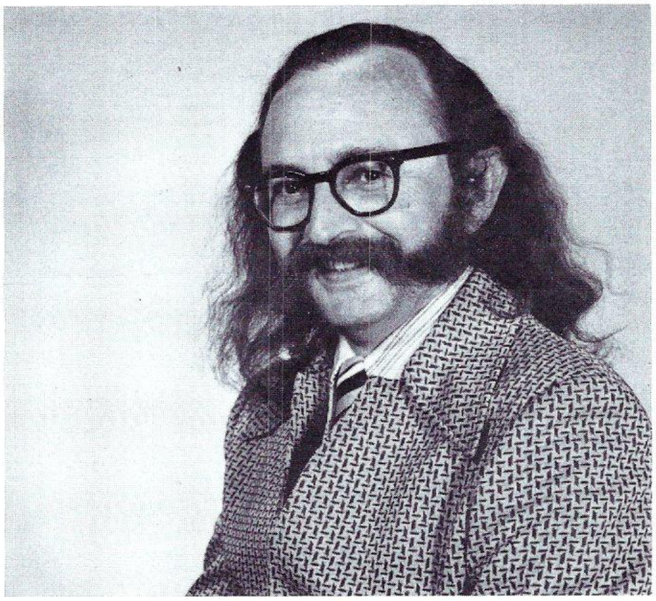
I1

CIRCLE NO. 7 ON INQUIRY CARD

Join the Club?

I read a very interesting editorial by Carl Helmers in the February issue of BYTE Magazine. Its headline is very similar to this one. As Mr. Helmers described a national personal computer society I started reaching for my checkbook and a self-addressed stamped envelope. I don't care what it costs, I want to join. But in the last chapter, like Don Quixote, the Knight of the Mirrors brings reality home with a vengeance. The enemy — in all its glory — turns out to be us.

My interpretation of Mr. Helmers' closing remarks is: if it looks like a duck, walks like a duck, quacks like a duck, it is most certainly a duck even if it calls itself the Southern California Computer Society. Sadly I gave up the vision of the Great Galactic Computer Cavalry riding to our rescue and went to the 942nd daily meeting of The Trivia and Tedium Standards Committee. The world has again proven that if you glue enough hang-nails together you get Frankenstein.



The stark naked truth is that we have no plausible defense against Mr. Helmers' charges. In fact, we are lucky he didn't expand the indictment to include "International Society", which we cannot deny either. I think it is unfair that he did not include a few mitigating facts, such as our 370-158 group purchase fiasco and the total lack of attendance at our Lunar Chapter meetings. With just a few more well planned, superbly executed disasters we might have escaped this one.

The dastardly attack on the Society name seems strange and somehow provincial from an organization aspiring to nationwide scope. At the venerable age of four months, the acronym SCCS is at least as recognizable as SPQR which was a fairly active society in its day. If our name is objectionable—so be it. We do not aspire to The Kingdon, The Power, and The Glory. We just hope to help the next guy who wants to know which is pin 1.

Our first and foremost goal in our charter is to provide the best possible service to anyone who asks. But we demand a heavy fee: the same caliber of service must be given to the next one who asks. On this premise and this name our organization stands, and all are welcome to join. There is no residence requirement, nor any other impediment. We are not the new Genghis Khan astride a forty legged horse called 8080 sweeping out of the west to cut the Gordian Knots of Computerdom. We are building the tools to pry open Pandora's Box and show computers for what they are and what they might become.

The name of our publication, INTERFACE, is probably also misunderstood. The computer interface is usually understood to be the common boundary between two subsystems. If the computer interface is proper the boundary disappears, and the whole becomes greater than the sum of the parts. This is why our INTERFACE has become our flagship. We have a message, we have a voice, we will speak, and we will be heard.

Lest anyone mistake our clarity of purpose for ambition, note that we have not asked Jeff Landry to change his name to Jeff Lendry to preserve our image. Also note that the illustrious founder of INTERFACE, Jon Walden, must look down from the Pantheon at the January issue where his name is wrong three times in three tries. The bell tolls for you Art Childs.

Returning to this scurrilous attack from BYTE; beware, Carl Helmers, the august president of SCCS is not powerless. One more outburst of this nature and we shall resort to the violence vested in me and name you honorary member and chairman of the colossal Name Committee (and publish your secret unlisted phone number. We haven't been infiltrated by the CIA for nothing!) On guard, sir, defend yourself!

If we can get back to our real business, we have a lot of work to do. We are building one brick at a time and we are competent builders. If anyone or any group wants to help—we need them. If help is needed, we will continue to do our best.

Respectfully submitted,
Ward Spaniol
President

A Lesson in Economics

What happens when a good becomes so desirable and sought after that many people want to have it? Unless an ample supply is forthcoming, the price can and must go up (political fiat to the contrary notwithstanding). Recently we wondered why one distributor kept asking us for more and more of BYTE Number 1. He was very insistent, and spent several transcontinental telephone calls urging us (in vain) to send him more of BYTE's inaugural issue.

We found out recently, thanks to several callers, that this fellow was selling BYTE's Number 1 issue for $15.00 each, since everybody wants one and the supply is very limited. Of course he made a temporary windfall profit on his early investment in a supply of the new magazine. He paid what everyone else did who purchased a bulk shipment of the then unknown magazine. There is a law of economics that price rises when supplies are tight – it is as much a folly to fight such a trend as it is to attempt to legislate Pi = 3.0000... Many thanks to Adam Smith and Ludwig von Mises for tipping us off about the law of supply and demand, confirmed again by this laboratory experiment.

The point of mentioning this is to put to rest rumors about our supposed reprinting of the early issues of BYTE. First, we are not planning to reprint BYTE magazine issues as magazines, ever. You'll see individual articles reappearing in books of selected reprints which we plan to publish. However our early supporters will see their confidence rewarded as the price of the first issue collector's item rises over the years, since we will never reprint it as a magazine. Our policy is now and will continue to be one of matching our press runs to the subscription and newstand demand, with very few extra issues kept available. If you want to keep up with this fast moving field, you'll have to keep your subscription current.

Cut

Name ....
Address ....
City .... State .... Zip ....
☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # ....

Tear

Name ....
Address ....
City .... State .... Zip ....
☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # ....

Shred

Name ....
Address ....
City .... State .... Zip ....
☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # ....

Rip

Name ....
Address ....
City .... State .... Zip ....
☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # ....

• BYTE • 70 Main St • Peterborough NH 03458

CIRCLE NO. 8 ON INQUIRY CARD

APPLICATIONS EXCHANGE

by Larry Press

The intent of the Applications Exchange is twofold: to describe applications which INTERFACE readers are running and are interested in, and to act as a "switchboard" to help people meet others with similar interests. In that our circulation has risen considerably since last month, I would like to mention a couple of items from the last column again for new readers.

First, George Tate is putting together an extensive games library. If you have games which he can duplicate or if you'd like to get a copy of a game playing program, write George at 3544 Dahlia, Los Angeles 90026. If you have an interest in biorythms, Art Childs has a BASIC program to compute biorythmic cycles. You may write Art at 335 N. Adams, #210, Glendale, Calif. 91206. Finally, I would like to gather information on two applications: bio-feedback and smoke detection. Let me hear from you if you have experience or interest in either of these areas. My address is 128 Park Place, Venice, Calif. 90291.

Future editions of the Applications Exchange will contain a directory of various applications and the people who are informally coordinating programs and information concerning them. Please let me know which applications you are interested in and whether or not you would like to be a low-key coordinator for any of them.

I haven't received any new Applications Exchange input since the February column was written, so I'm going to digress this month and discuss *tutorials* rather than applications.

In short I would like to encourage readers of INTERFACE to submit tutorial articles and I offer here a few suggestions for subjects. We need tutorial articles because SCCS has an heterogeneous membership—some members have backgrounds in hardware, some in software and a few in both.

My background is in software, so let me describe the sorts of hardware tutorials which would be helpful to me (someone else can do the same for software.) In my initial efforts to learn something about hardware, I have discovered three subjects of which I know very little: electronics, electronic technique, and digital design.

If you are up for writing a tutorial article on some aspect of electronics, bear this in mind: a person may be a professional programmer, systems analyst, user/or salesman, without having to know a volt from an ohm; the converse is not true. Because a hardware expert might inadvertently overestimate the degree of sophistication of the reader of a tutorial, articles on elementary topics such as "analysis of simple DC circuits", "what is a resistor/capacitor/diode/transistor, and how are they used", "what is a power supply", "how do power supplies work", "what goes on inside a TTL NAND gate", and others would be of value.

We could also use tutorials on electronic technique. Last week I decided to put together my first electronics project. I bought a printed circuit board and some components in order to make a 5 volt power supply. I'd read an article on power supplies and understood the circuit fairly well, but when confronted with the concrete parts, rather than those nice abstract symbols on a sheet of paper, I realized that I needed some nuts and bolts information. My first step was to call Pearce Young and ask him how to solder! Once he explained that little detail, I proceeded, and soon discovered that making the case to hold the power supply was a bigger job than putting the power supply together. Nuts and bolts tutorials on such topics as "how to solder", "how to etch printed circuit boards", "wire wrap vs. printed circuits vs. ?", "handy tools for electronic construction and assembly", "how to use VOM", "what are all those funny knobs doing on my oscilloscope", "the ins and outs of buying surplus components" . . . any and all would be appreciated by the hardware beginner.

The final subject is digital design. Of the three areas under discussion, this is where the software person is likely to be off to the best start in that he is familiar with concepts such as bit, memory, register, counter, encoding, state, and algorithm. There is even a good chance that he has seen some formal logic in school. He would still have much to learn from tutorials on topics such as "what is an IC", "What are the similarities and differences between TTL, DTL, etc", "simple experiments to try with a NAND chip", "noise filtering in IC projects", "flip-flop chips", "basic gates". You experts and novices can surely think of topics to share or inquire about.

Well now, why don't you think about writing a tutorial article for INTERFACE? If you are an expert in one of these areas, beware of the pitfall of assuming that the reader has acquired concepts which you take for granted (mysterious terms like "PC board" and "IC socket"). Be sure that you define terms and provide references to prerequisite articles. On the other hand, if you are a beginner, you won't have this

(Continued on page 48)

Bits & Bytes

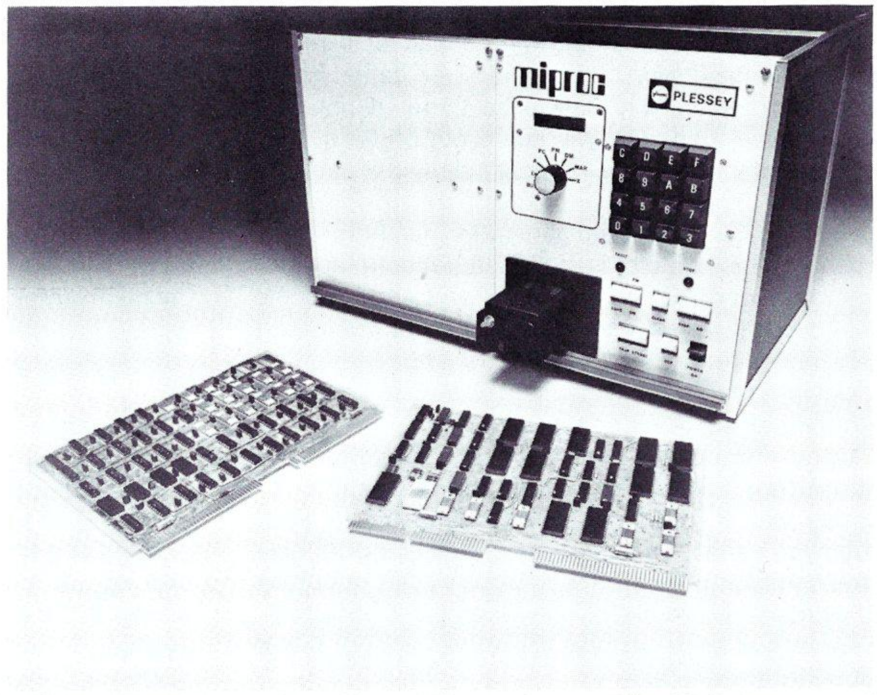
by Mike Teener

Well, it's still not in . . . my ALTAIR 680, that is. MITS has recalled all 680's before they were delivered to install a couple of new features. First, an ACIA in the serial interface (this is a single chip programmable serial interface—sort of like the 8800's, 2510 board), instead of a true bitserial interface. I appreciate that, MITS; I like to do nitty gritty level programing, but to have to output one bit at a time is just too much!

A 256 Byte PROM with a basic loader/bootstrap program is going to be included this time, and just to keep my spirits up, they are going to give me a 10% discount on the 8K memory board when it comes out. Good for you guys. You may be slow, but at least you give me something for it.

Also from MITS—there will be a new version of the 8800 with improvements to make it more convenient to build and operate. No details, though.

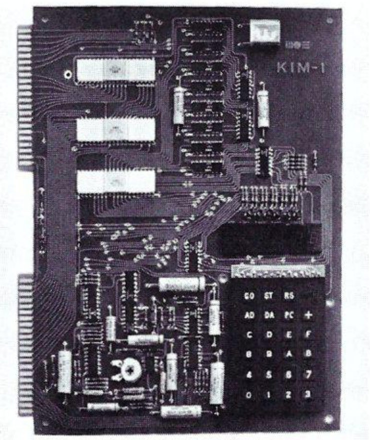
Here's an interesting microcomputer for you from England. The Plessey MIPROC . . . 16 bits, 350 nanosecond instruction execution time! Good God Almighty! Data and program memory are separate. You can have up to 65K words of both data and program, if a little crude. It only has four registers, with the program counter and memory address register included. It looks like a strange combination of an HP 2100 mini and an 8080. Plessey Microsystems has an office in Southern California at 1674 McGraw Ave., Irvine 92714.



Plessey's MIPROC Microcomputer contains 16 bits, 350 nanosecond instruction execution time, with separate data and program memory.

Photo supplies by Plessey

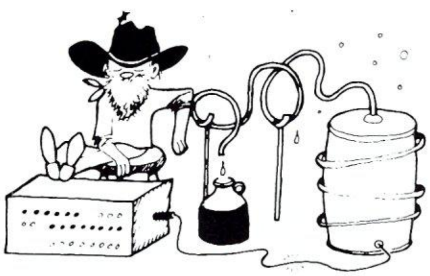
Another new microcomputer. MOS Technology (6502 and 6501) has built a complete microcomputer called the KIM-1 which puts the JOLT to shame. I told you about the KIM version of the 6530 chip three months ago. Well, here it is in hardware form—complete with HEX keyboard and display, ROM monitor, TTY and audio cassette interface, and the best programing and hardware manuals in the business. Cost? $245 assembled. You add a power supply. Their address is 950 Rittenhouse Road, Norristown, Pennsylvania 19401.



MOS Technology Model KIM-1

BLUE SKY DEPARTMENT. I just finished a pilot project at work on computer word recognition. I got it working well enough to recognize spoken digits with over 95% accuracy. The problem turned out to be a lot easier than I feared at first . . . as a matter of fact, I think I could get it going on a microcomputer without too much trouble. Combine both linear and digital LSI technology, and it would be possible to build a new I/O board for your computer . . . spoken words . . . inside ten years at the very longest.

GAMES & THINGS



BY Tom Rugg

In nearly all game playing programs, there is the need for some randomness. After all, if you play a game against a computer that does exactly the same thing every time, you're bound to get bored with it before long. So, whether it's shuffling a deck of cards in a blackjack game, or throwing a pair of dice, or "setting up the universe" in a Star Trek game, some kind of unpredictability has to be there. If not, the human player generally will figure out pretty quickly how to beat it every time (or conclude that it can't be beaten).

A subroutine or program generating numbers which appear to be random is called a random number generator, or RNG. To be more precise, it should really be called a *pseudo*-random number generator, because it really only gives the *appearance* of randomness in a very systematic way. But we'll stick with the name RNG for simplicity.

Many computer systems and compilers have pre-written RNGs. But maybe yours doesn't, or maybe you'd like to have one tailored to your particular needs. In any event, we'll look at how three common types of RNGs can be used, and then we'll go into the details of how to program your own.

For lack of better names, the three types of RNGs will be referred to as:

1. the "fixed range integer" type
2. the "variable range integer" type
3. the "zero to one floating point" type

Let's see how each of these would be used if we needed to throw a pair of dice in a program. What we need are a pair of random integers, each in the range of 1 through 6.

The "fixed range integer" type will provide a random integer within a given range, such as zero to 1,000,000. Probably the simplest way to convert this random number to a random integer in the range from 1 to 6 is to divide it by 6 and add 1 to the remainder (*not* the quotient). If we call the RNG twice and do this each time, we have our pair of dice.

The "variable range integer" type is even easier to use, because it takes care of the conversion for us. When we call it, we specify that we want an integer in the range of 1 to 6, and the RNG provides exactly that. So, all that is needed is to call the RNG twice, each time telling it that the desired random number range is 1 to 6.

The "zero to one floating point" type is probably the most common general purpose type of RNG, but is a little more awkward to use for the dice requirement. When called, it will return a random floating point number in the range of zero to 1 (actually this type generally uses the range of zero to .9999. . .). To get our integer from 1 to 6, we can do one of the following:

1. Assign the appropriate integer based on which sixth of the range the random number falls in (e.g., if between zero and .1666, set it to 1; if between .1667 and .3333, set it to 2, etc.).
2. Multiply the random number by 6, add 1, and use the integer portion of the result.

PROGRAMMING A "FIXED RANGE INTEGER" RNG

The first type of RNG is pretty easy to program, and should be especially useful on a low cost microcomputer because it doesn't require any multiplying or dividing. This algorithm is similar to one presented in *Problems for Computer Solution* by Gruenberger and Jaffray which, by the way, has an excellent discussion of RNGs and how to test their "randomness".

At the heart of this RNG (and most others) is modular arithmetic. Even if you're not a mathematician, you already have a great deal of experience with modular arithmetic, although you may not realize it.

For example, what mileage will your car's odometer show 20 miles after it shows 99995 (disregarding tenths)? It won't say 100015, because it only goes up to 99999, so it will say 00015 (unless it's a Volvo which turns over at a million). In other words, the typical car odometer does modular arithmetic using a *modulus* of 100,000. Just as the car's odometer limits its numbers to the range zero to 99999, our RNG will limit its numbers to the range of zero to one less than some modulus we use, which is based generally on the word size of the machine we're computing with.

As the flowchart in figure 1 shows, we will double the number K and reduce it by the modulus MODK. Similarly, we'll double the number L and reduce it by the modulus MODL. Then we'll add K and L to M, which will be reduced (more than once, if necessary) by MODM. M will then be our random number in the range from zero to (MODM-1).

Although there are many ways to decide what numbers to use for MODK, MODL, and MODM, it is best if they are prime numbers of the form (8n + 3) or (8n + 5). In addition, it is generally best if all three are slightly less than one-third the maximum number that will fit in a word. For example, 32,767 is the maximum for a 16 bit word machine (assuming one bit for the sign), so use three numbers that are less than 10922. Suggested values are 10909, 10891, and 10861.

A problem that needs to be solved is that of the starting values for K, L, and M. First of all, they must be in the range of 1 to one less than the appropriate modulus. Additionally, it is generally best if the three starting values are not too close together.

But, we can't initialize these variables to the same set of values every time we first call the RNG in a game, or else we will get the same sequence of random numbers each time. What is needed is an unpredictable "seed" to start it with. There are several ways this can be done:

1. Ask the human game player to provide a starting value within a specified range.
2. Use some unpredictable variable within the computer itself, such as the time of day in seconds. It may require manipulation to put it within the range of numbers you need.
3. Build a "perpetual" RNG into your system that exists from run to run, using the last random number from its previous run as the seed for the next run.

Each method has advantages and disadvantages, but getting a starting value from the player can be especially useful during testing of your game programs because, for example, you can duplicate an earlier test after having fixed a bug.

Keep in mind that you don't need three unpredictable seeds (for K, L, and M). You can program K and L to always start at 2000 and 7000, for example, and then just get a seed for M.

In future issues, we'll look at other random number generators and the ways they can be used in game playing programs. If you have a favorite RNG and/or a clever way of using one, send it in. And if you have a favorite game, please send it in, too. Address all correspondence to the Games Editor, in care of *Interface*. ▣

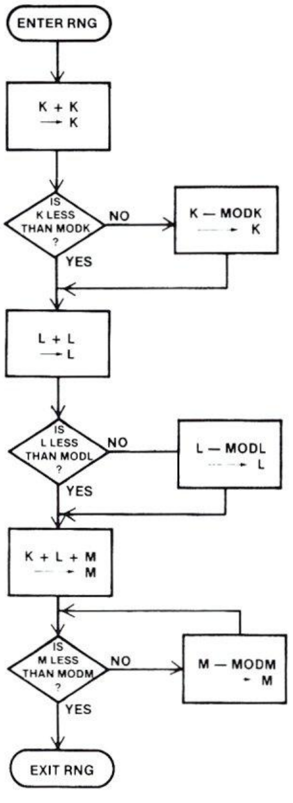
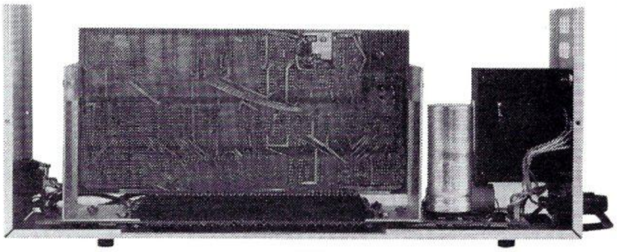


Figure 1

INTRODUCING THE

MICRO-ALTAIR™



The complete Computer System that requires just a keyboard and TV monitor for use.

The MICRO-ALTAIR™ consists of our video board with graphics capability, CPU/ROM/RAM board, backplane with power supply, and cabinet.

The CPU board includes an 8080 processor, 512 bytes of RAM, space for 3K bytes of ROM and vectored interrupts. Several CPU boards may be plugged into the same backplane for parallel processing.

The backplane design is unique in that it allows many backplanes to be plugged together for easy system expansion. Each backplane contains its own power supply (transformer mounted externally) rated at 8 volts at 6 amps and ±18 volts at ½ amp. 5 boards may mount in each backplane assembly.

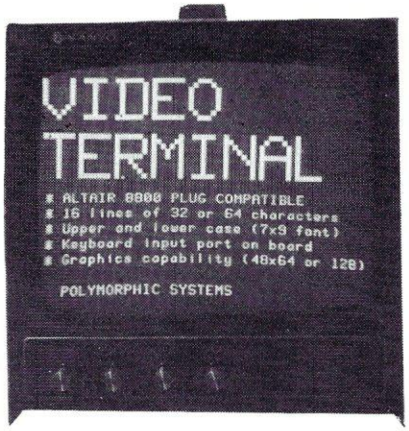
Included with each system is a resident operating system contained on a PROM which plugs into the CPU board. The operating system implements a versatile file system, provides program debugging aids, handles input and output concurrently with program execution, provides job scheduling and a real-time clock. Data may be entered and listed in octal, hexidecimal, or ASCII and edited on the TV screen. Files may be read from or written onto external devices such as a cassette tape. Program debugging aids include software breakpoints which allow all CPU registers to be displayed at any point in the program execution.

An important feature of the MICRO-ALTAIR™ is its compatibility with Altair peripherals and software which are available from several manufacturers.

PRICES:

Cabinet and power supply	$ 155.00
Processor board (includes RAM, real time clock and vectored interrupt)	195.00
Video interface board (with 8-bit input port)	160.00
Operating system on PROM	65.00
2 100 pin connectors	17.00
The complete system purchased separately	$ 592.00
SPECIAL: INTRODUCTORY OFFER	**$ 475.00**

This offer applies only to orders accompanied by check or money order and postmarked before April 15, 1976. We believe this is one of the best deals going in a computer kit, check our prices and compare. Delivery 60 days.


VIDEO
TERMINAL
* ALTAIR 8800 PLUG COMPATIBLE
* 16 lines of 32 or 64 characters
* Upper and lower case (7x9 font)
* Keyboard input port on board
* Graphics capability (48x64 or 128)
POLYMORPHIC SYSTEMS

Video Interface. Plugs into the Altair or IMSAI 8800 bus and connects to standard TV monitor for modified receiver). 8-bit input connects to almost any keyboard. Characters are displayed as 16 lines of 64 or 32 characters each, in a 7x9 dot matrix. (64 character line requires a high resolution monitor or extensively modified receiver.) The character set includes 128 upper and lower case ASCII characters and 64 graphics characters for plotting a 48x128 (or 64) grid. Characters are stored in the onboard memory and may be accessed directly by the computer.

$160 Kit $230 Assem. and tested (Add $25 for 64 character option) delivery 30 days

Analog Interface. This board is a complete interface for a CRT graphics display or an X-Y plotter, and provides 8 channels of software-controlled A/D conversion. There are 1 or 2 channels of analog with 10-bits resolution (0-10v or ±5v out), 6-bits of latched digital output, and 8 analog comparators.

1 channel (0-10v): $135 Kit $175 Assem. and tested
2 channels (0-10v): $185 Kit $235 Assem. and tested
(Bipolar option (±5v) add $8 for 1 ch., $12 for 2) delivery 30 days

All prices and specifications subject to change without notice. Prices are USA only - Calif. res. add 6% sales tax.

POLYMORPHIC SYSTEMS
737 S. Kellogg, Goleta, CA 93017 (805) 967-2351

BankAmericard and Master Charge Accepted

CIRCLE NO. 9 ON INQUIRY CARD

HARDWARE REPORT

ESSENTIALS FOR THE BEGINNER

S. Wilcox

The modern computer is a tremendously complex machine. Direct exposure to all its features can be more bewildering then inspirational. When one starts to use a computer, he has an immense array of programs to control his access to it, to help him avoid errors, and to provide tools for creating MASSIVE programs when all that he really wants are simple ones. There are myriad conventions to memorize and dictionaries of diagnostic messages to help him out of any error situations. Needless to say, it takes a while to settle in.

IN THE BEGINNING:

There was a time when machines had a memory capacity of no more than a few thousand words, a minimal set of arithmetic special registers just adequate to carry out computation, a set of instructions just adequate for arithmetic and stored program control, and a control unit to sequence program execution. There was one input device; punched card, paper tape, or magnetic tape, and one output device which may have been an electric typewriter, magnetic tape recorder, or a card punch. That was the era of the von Neuman machine. (How times have changed.)

It would be difficult to recreate the days of the von Neuman machine and as revolutionaly as it was, some of its headaches might be as well avoided. However, the sheer beauty of working with a simple machine architecture, an uncluttered instruction set and no supervisor watching over us to recount our mistakes can be a joy of its own.

THE INITIAL (IMAGINARY) MACHINE

MEMORY: The computer has a internal memory of 1024 thirty-two bit binary words. Negative numbers are stored in two's complement form. Each word is subdivided into 4 eight bit parts, each part called a byte. Individual bytes are addressable, the addresses ranging from 0000 to 4095.

ARITHMETIC REGISTERS: There are two registers which take part in arithemetic (as opposed to control). Each register has 32 bits. They are known as the A and B registers. Multiplication and division require a 64 bit register formed by the coupling of the A and B registers. Addition and subtraction take place in the A register, also known as the accumulator. Arithmetic shifts provide high speed multiplication and division by a power of two as applied to the full 64 bit coupled register pair AB. Logical shifts are used to facilitate operations with logical portions of words.

CONTROL REGISTERS: Each instruction occupies one full word and the normal successor to the instruction with address N has the address N + 4 for this machine. The instruction sequencing is controlled by the Instruction Location Counter which is inaccessible to the programmer although it can be modified by some of his instructions. The Instruction Location Counter has N as its contents when the instruction with address N is brought from memory and placed in the Current Instruction Register for decoding. The Instruction Location Counter, also known as the Program Counter, is incremented by 4 before the next instruction is needed.

If a branch is made to an instruction sequence elsewhere in memory — call it the target sequence—the incremented address in the Program Counter will be replaced by the address of the first instruction in the target sequence. Before this occurs, it may be necessary to establish an address to which a return may be made from the target sequence when it is completed. This return address is saved by loading it into a holding register before branching.

INSTRUCTIONS: An instruction is one word in length. It is written in hexadecimal (base 16), two hex digits to each byte, or in octal (base 8), three digits to the byte. Each instruction has its leftmost byte devoted to an operation code (OP Code). This code determines the role of the remaining bytes of the instruction. These remaining bytes may name a register, constitute a memory address, or be data as the argument of the OP code.

Do any of these essentials sound familiar? They should if you have taken a course in computer science and should be even more so if you own your own computer. Do not let yourself be blinded to the essentials by the growing complexity of your system's hardware and software. There is an elegance in simplicity.

HARDWARE FOLLOWUP

In the December issue of INTERFACE it was presented that the 4K memory board offered through Godbout Electronics had a power consumption of 1.7 amps. Shortly after the December issue was released, Bill Godbout called to tell us that something was definitely wrong with that particular memory board. There now appears to be some error regarding the meter used.

After re-evaluation on different meters, the real current draw should be in the range of 1.1 amps. The original measurement was done with an EICO model 565 VOM which has a stated accuracy of **+3%** of **Full scale.** The limiting error on the 10 amp scale which was necessary for this measurement is therefore 300 milliamps. The fractional limiting error for a 1.7 amp reading on this scale is therefore

$$\%Qe = (.3/1.7)100$$
$$= 17.6\ \%\text{error}$$

Errors of this magnitude are unacceptable for all but "ballpark" figures. The best that should have been said was that the Godbout 4K memory draws slightly over 1 amp. Our apologies to Bill Godbout and our thanks for his cooperation in this evaluation.

HARDWARE SURVEY

In order to better understand our growing field and serve the general thrust of interests, we would like to know about the personal computing systems that our readers have built or otherwise collected into an operational system (or potentially operational if yours is not 'up' at this time).

We know that there are many types of systems in existance and do not want to give the impression that the realm of home computing is owned by ALTAIR. There is club interest in all types of systems at various levels, their aspects and applications. Please let us know what you are doing with your system and what hardware you are using to do it.

Survey Form

(Continued on page 48)

POWER-ON-CLEAR VS. PROTECT

S. Wilcox

There is a corollary to Murphy's Law that says the hassels you experience with your system are directly proportional to the complexity of it. But, hopefully, we learn in the process and tune ourselves for minimum headache pain.

How many of you have powered up, hit your bootstrap and started playing in the mag or paper tape with the happy thought of soon giving the Klingon flagship a broadside of photon torpedos only to find later that all was for naught because you inadvertently tried to load the program into a protected block of memory?

As a matter of convenience all volatile RAM should come up in the unprotected state upon application of power. Otherwise you must examine an address on each memory board in your system to see how it came up. Some memory boards being sold have provision for an automatic power-on-clear for the protect/unprotect flip-flop, others do not. Usually a particular memory board without POC (power-on-clear) will favor coming up in the same state each time power is applied.

It is possible to force boards without POC to always start in the unprotected mode. If your system's buss structure has a power on clear signal you may take advantage of it, or you can add your own. Most memory protect circuits use some form of gated flip-flop as shown in Fig. 1. Add the 1N914 diode as shown if POC signal is available. If no POC signal is available, add the circuit of Fig. 2 to each memory board or to your buss structure.

The only requirement is that the power on clear signal be present for several milliseconds as the power supply voltage is building up, to bias the flip-flop to the unprotected state. ◼

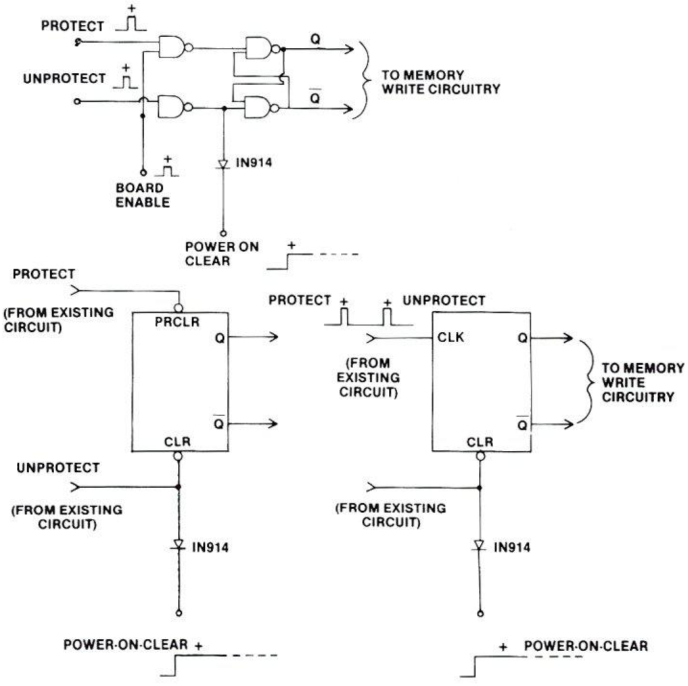

PROTECT
UNPROTECT
Q
Q̄
TO MEMORY WRITE CIRCUITRY
BOARD ENABLE
IN914
POWER ON CLEAR
PROTECT
(FROM EXISTING CIRCUIT)
PRCLR
Q
Q̄
CLR
UNPROTECT
(FROM EXISTING CIRCUIT)
IN914
POWER-ON-CLEAR
PROTECT
UNPROTECT
CLK
(FROM EXISTING CIRCUIT)
Q
Q̄
CLR
TO MEMORY WRITE CIRCUITRY
(FROM EXISTING CIRCUIT)
IN914
POWER-ON-CLEAR

FIGURE 1

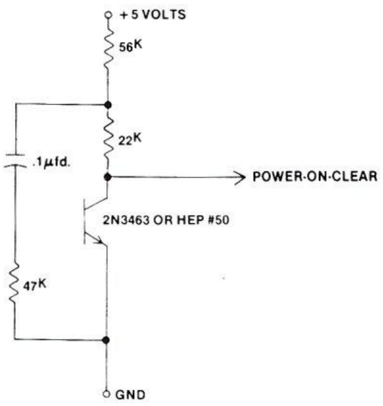

+5 VOLTS
56K
22K
.1µfd.
POWER-ON-CLEAR
2N3463 OR HEP #50
47K
GND

FIGURE 2

Letters to the Editor

The editor of INTERFACE Magazine wants to know how YOU feel, what YOU think, and respond to your suggestions and requests. Write Editor, INTERFACE Magazine, 335 N. Adams St., Suite 210, Glendale, Calif. 91206.

John Borders contributed the construction details for a small, but nice addition to the Altair 8800. The last paragraph of his accompanying letter read:

I hope you don't envision this taking up very much space in the INTERFACE. While I do feel that my Altair Aux is a worthwhile addition to any Altair, I don't feel that it compares with Ward Spaniol explaining why the SCCS has a chapter in the Delaware Valley.

John Borders
Oxnard, Calif.

Dear John:

You'll find Ward's "explanation" in this issue on page (36). Readers can look forward to your contribution in next month's INTERFACE.

Editor

Dear Editor;

All of us in the Homebrew Computer Club think you people are doing a great job. The new issue of *Interface* is very professional. I hope we can work together . . . Maybe several newsletters could all have space . . . If you are interested, I will support the idea here.

Ken McGinnis
San Mateo, California

Dear Ken:

Thank you for your kind words. Your proposal is worthy of consideration. Material from other clubs is very welcomed since it would increase the range of *Interface* editorial content. SCCS certainly has no monopoly on talent, and we're dedicated to doing all for the home computerist that this magazine can accomplish.

Editor

Dear Editor;

The January issue of BYTE Magazine contained an advertisement which posed an interesting question, "Who should join SCCS?". After finding that the three conditions which accompanied the question applied to myself, I've decided to ask you a question. "May I become a member of society?"

Truthfully what really prompted me to ask my question was not your advertisement, but rather the enthusiasm I encountered one evening in a late night phone conversation . . . We discussed the . . . quantity purchase and other club activities. I finished the conversation with a very positive feeling that your group could help me, the seemingly alone hobbyist.

Bill Kittle
Ann Arbor, Michigan

Dear Bill:

Don't feel alone. True, many hobbysits have found a home in the Southern California Computer Society, but don't let our name deceive you. You have friends all over the USA and abroad who are members and are forming chapters. Won't some member in the Michigan area come forth in brotherhood to prevent invaluable computer talent from going underground?

Editor

I was intrigued when I found my name mentioned in "Interfacial" referencing a letter that apparently didn't fit into the "Letters" department! But never mind, there's no hard feelings. It's hard to see how such a fine publication can appear with so few hands and then on a part-time basis.

I was a little unsure where to send editorial material. However, I was able to find your address in Issue 1. I would suggest giving explicit information to prospective authors on the inside front cover or other prominent place. This should give you more material from which to select during this critical stage in a new publication. If you later need help in reviewing or refereeing papers, I could help in the areas of the Altair 8800, programming languages, and large-scale computers.

Richard R. Kenyon
Cerritos, Calif.

You won't have to look far to discover that we've taken your suggestions seriously. Thank you for the compliment, and your offer to help, Dick. We accept.

Editor

Dear Editor,

I would like to submit two programs (8080) that the beginner programmer (like myself) might be interested in. The first program moves a block of data from one location to another location. Notations are in octal. The program can be placed in any location in memory. The block of data can be any Number of bytes with the exception that N $\frac{2^{16} - 26}{2}$.

The second program compares one block of data to another block of data. The above conditions apply with the exception that N $\frac{2^{16} - 32}{2}$. Should the program branch to the error loop, interrogation of the HL registers will reveal the error.

ROUTINE TO MOVE BLOCK **A** TO BLOCK **B**

```
ADDRESS8  CODE8                COMMENTS
000 000    041    LXI HL     Block "A"
    001    000    B2  Lo     Begin Address
    002    000    B3  Hi
    003    021    LXI DE     Block "B"
    004    000    B2  Lo     Begin Address
    005    000    B3  Hi
    006    001    LXI BC     Set counter to N bytes
    007    000    B2  Lo     to be moved
    010    000    B3  Hi     from location "A" to "B"
    011  A 176    MOV A,M
    012    022    STAX DE    A to M addressed by DE
    013    023    INX DE     by 1
    014    043    INX HL     by 1
    015    013    DCX BC     by 1
    016    227    SUB A      Clear accumulator
```

MOVE BLOCK **A** TO BLOCK **B**

```
ADDRESS8  CODE8                  COMMENTS
000 017    270    CMP B with A
    020    302    JNZ          Jump of /=      to "A"
    021    011    B2  Lo
    022    000    B3  Hi
    023    271    CMP C with A
    024    302    JNZ          Jump if /=      to "A"
    025    011    B2  Lo
    026    000    B3  Hi
    027    303    JMP          Wait here when done
    030    027    B2  Lo
    031    000    B3  Hi
```

ROUTINE TO COMPARE BLOCK **A** TO BLOCK **B**

```
ADDRESS8  CODE8                COMMENTS
000 000    041    LXI HL     Block "A"
    001    000    B2  Lo     Begin Address
    002    000    B3  Hi
    003    021    LXI DE     Block "B"
    004    000    B2  Lo     Begin Address
    005    000    B3  Hi
    006    001    LXI BC     Set counter to N bytes
     07    000    B2  Lo     to be compared
    010    000    B3  Hi
    011  B 032    LDAX DE    Load A from M addressed by DE
    012    226    SUB M
    013    302    JNZ        Jump if /=      to C    error
    014    035    B2  Lo
    015    000    B3  Hi
```

COMPARE BLOCK **A** TO BLOCK **B**

```
ADDRESS8  CODE8                COMMENTS
000 016    023    INX DE     by 1
    017    043    INX HL     by 1
    020    013    DCX BC     by 1
    021    227    SUB A      Clear Acc.
    022    270    CMP B
    023    302    JNZ        Jump if /=      to "B"
    024    011    B2  Lo
    025    000    B3  Hi
    026    271    CMP C
    027    302    JNZ        Jump if /=      to "B"
    030    011    B2  Lo
    031    000    B3  Hi
    032    303    JMP        Wait here if A = B
    033    032    B2  Lo
    034    000    B3  Hi
    035    303    JMP        Wait if Error
    036    035    B2  Lo
    037    000    B3  Hi
```

SOFTWARE DOCUMENTATION STANDARDS

By Bill Roch

Most people in the home computing field are non-conformists in one way or another —why else would we expend the time and money to have at home what many of us work with all day long. So it seems contrary to our purpose—promoting and supporting home computing—to talk about conformity, but our computers are constructed from standard parts, and demand a certain degree of conformity to operate. (Imagine trying to build your Sphere System 4 with resistors from three or four different manufacturers, each of whom used his own color coding scheme.) Further, to utilize our systems with the greatest ease possible, some standardization in the area of software is going to be necessary. The general item "software" implies several categories of media requiring imminent attention: cassette tape, floppy disk, paper tape, telephone transmission, etc.

When considering these categories, many questions arise. Do we use ASCII or a different code? Do we standardize on header and trailer label formats, and if so, what data should be contained in the headers?

Documentation is another consideration, and has been chosen as a starting point for discussion. The ideas presented here are intended to act as a stimulus to begin documentation standardization.

Now that an audio cassette interface (modified Lancaster) has been adopted we have a partial means of swapping software. Perhaps the next thing to standardize is the software we swap. To this end we are suggesting the following standards for consideration to use with public domain software. These standards may not be the ultimate by they do provide a base from which to change.

SWAPPING CONCEPT

Each club could accept software from its members that meets standard and make it available to all its members and other groups on cassette tape (documentation should be on cassette, if possible) on a trade or nominal charge basis. Each club would be responsible *for* issueing a sorted catalog of their software periodically. One or more of the clubs would maintain a master catalog for issue to all clubs periodically.

This concept is no way intended to restrict individual swapping. If Joe and Mike want to swap software—fine—more power to them. It should, however, help restrict the exchange of undocumented non-standard software which winds up in a bottom drawer because it is too much trouble to rewrite or decipher.

ANSI & EIA

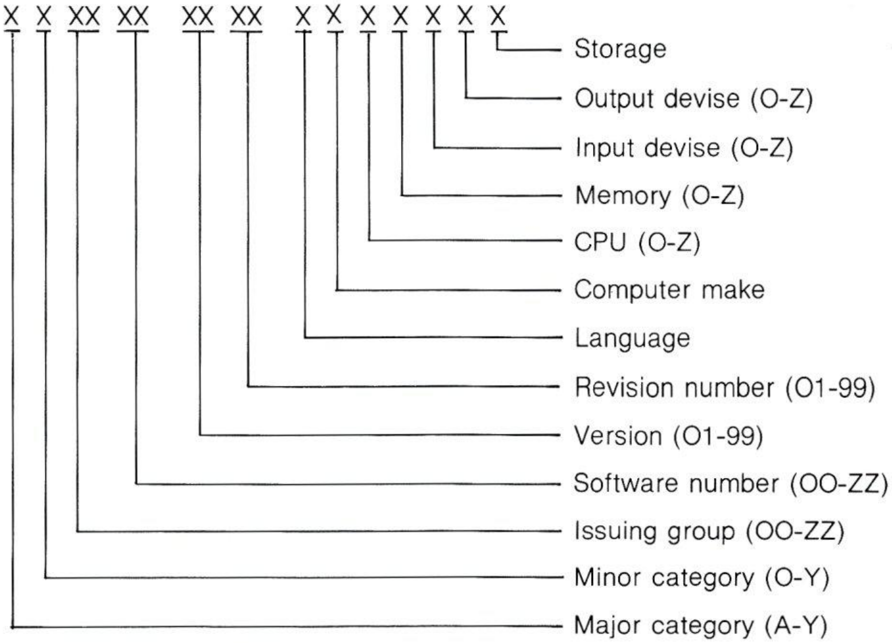
ANSI and EIA standards should be followed wherever practical.

CATALOGING

Many individuals have "pride of authorship" and would rather "reinvent the wheel" than use software already written. However, most of us would be happy to use existing software if:

- We know it exists
- We know where to obtain it
- It does not cost an arm and leg
- It has good documentation

We first need a catalog of software which means program identification. Since floppy disks will be coming into use soon with a probable six-character file name, the following catalog numbering system is proposed:



Category ZZ would be reserved for uncatalogued use.

EXAMPLE: GGSC09 — 08.02 — BA83CCF

Games, guessing type, Southern California Computer Society — the eighth version of that game, revision 2 in BASIC for an Altair with 8080 CPU chip, requiring three 4K memory boards and uses CRT keyboard input and CRT output running from a floppy disk.

The 72 character catalog listing for this example follows:

GGSCO9 — 08.02 GUESS A WORD BA83CCF
LETTER BY LETTER WITH 200 WORD FILE

Using this coding scheme, it would be possible to select and sort the catalog by language, CPU, etc., and combinations, such as file maintenance programs in FORTRAN for a floppy disk system. No attempt has yet been made to set up major and minor categories. Suggestions for categories from you are invited.

DOCUMENTATION

Everyone who writes software has his own level of documentation ranging from non to "over kill"; it is usually just a little more than none.

For someone else to make use of your work of art, he must know certain information about it, such as:

- Language (BASIC, Assembly, FORTRAN, etc.)
- Minimum machine configuration required.
- Purpose of description of what the software does.
- Valid answers to program questions.
- Error messages and their meanings.
- Input and output format requirements.
- Cassette tape format.

Pertinent information for your particular software.

Other items should also be included such as:

Software name.
Author's name.
Date.
Club name.

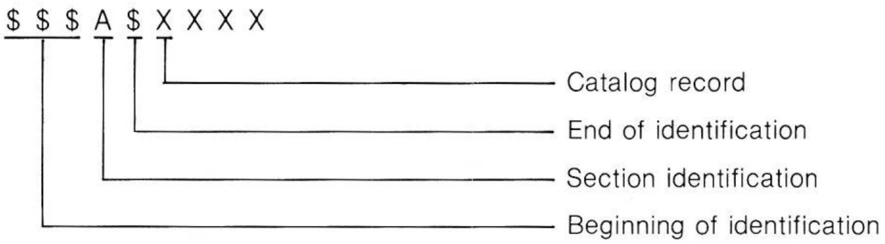
There is no one best way, since language and type of software varies. A little BASIC program can be handled by remarks ahead of the program. A complex program may require "x" teen pages. Assembly language programs should have both HEX and OCTAL notations together with comments.

Documentation that is not involved as part of the software program should be put on cassette together with a loader for reading the tape in, and a print program for the writeup. This keeps the swapping organization from becomming a "paper mill". If you don't have a printer, get your buddy to print out the documentation for you.

One possibility to consider is as follows:

Section I- Catalog record
Section II- Comments
Section III- Program
Section IV- Trailer

Each section would consist of an indentifier, perhaps three seldom used characters to indicate that the next character denotes the section.



With such a system, the operating system could be programmed to recognize and handle each section properly. For example, upon recognizing the catalog section, it could be displayed on the CRT, then wait to allow the operator to verify that the proper file has been accessed. Upon recognizing the comments section, the O.S. could check a switch previously set by the operator to determine the need to display or print the comments. the O.S. could check a switch previously set by the operator to determine the need to display or print the comments.

Upon recognizing the program section, the first record could be checked to determine load point, peripheral assignment, etc. By proper setting of sense or program switches, the operator would have the option of bypassing the first two sections provided he was sure of ID and familiar with the comments.

DOCUMENTATION DISPLAY FORMAT

Over a period of time the most useful documentation display format will emerge because it will be copied. Until then, it will be somewhat a "whatever" format.

Home computing is obviously here to stay and it's growing. Already some manufacturers are on the verge of hardware design decisions that could eventually lock us into a less than desireable standard, or worse, create a profusion of standards that would only lend confusion and frustration to home computing. To facilitate the "stitch in time", readers are invited to respond. The author will attempt to serve as a data collection function for all adeas on the subject and communicate the results in future issues of *Interface*. Your response may be made directly to Bill Roch, 5658 Keokuk Ave., Woodland Hills, California, 91364, or through INTERFACE Magazine. ◘

Applications Exchange Tutorial

(Continued from page 38)

problem. Tutorials written by beginners as they learn would be quite valuable. What is lacking in technical depth would be compensated for by sensitivity to the pitfalls and misconceptions of a fellow newcomer to the subject.

Next month I will continue to discuss tutorials along with the Applications Exchange. Included will be mention of some of the useful tutorial material which I have found elsewhere. If any of you are aware of good tutorial books or articles, I would appreciate your letting me know of them so I can pass them on next month. ◘

In This Issue

(Continued from page 4)

functions of editor and referee are needed—particularly from areas other than Los Angeles. We in Southern California sincerely want to know what's going on in other parts of the country (world?) so if you'd care to contribute rather than edit, we'll welcome you to our pages. Put that last statement in spades if you have 6800 software. ◘

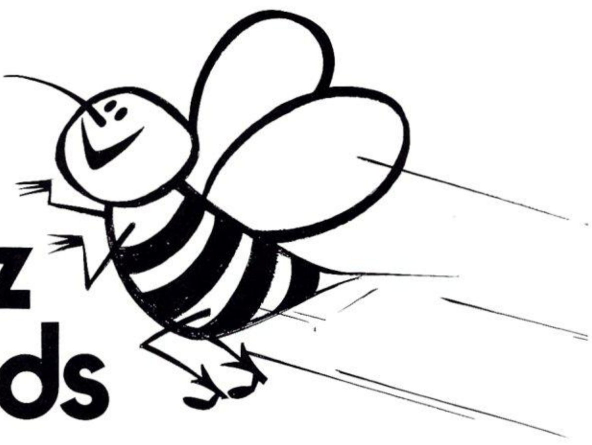
Hardware Survey Form

(Continued from page 43)

1. What CPU chip is your system based on? ________
2. Homebrew? ____________________ or commercial product? ______________________________ (trade name)
3. How much memory do you have? ________ K RAM ________ K ROM
4. What type software are you running? ____________ (assembly, BASIC, etc.)
5. What input/output devices do you have? ______________________________ (TV, printer, punch)
6. What are you using or plan to use your system for? ______________________________
7. What else are you just dying to tell us? ________ ______________________________

Feel free to photocopy this, or add extra pages as you require. Sign your name if you wish, and mail your response to: S. Wilcox, Hardware Editor, 405-B North Monterey, Alhambra, Calif. 91801. ◘

buzz words for the computer enthusiast



by Terry Benson
Field Applications Engineer, Intel Corp.

Use them in your everyday conversation—you'll sound like an applications engineer.

ALU — Arithmetic-Logic Unit. That part of *CPU* which executes adds, subtracts, shifts, and performs logical operations.

ASSEMBLER — *Software* that translates an *assembly language* program into a machine language (object) program.

ASSEMBLY LANGUAGE — A shorthand notation which saves the programmer the trouble of remembering the bit patterns of each machine language instruction. One machine language instruction is generated for each assembly language instruction.

BCD — Binary coded decimal (4 bits represent the digits to 9 only).

BUS — A group of lines which provide a common communication path between *CPU, MEMORY* and *I/O* devices.

BUS DRIVER — I.C. that buffers devices from the *CPU* and provides the capability of incorporating many additional devices in a *microcomputer* system.

COMPILER — *Software* to translate a program in a high-level language such as Fortran or *PL/M* into a machine language (*object*) program.

CPU — Central Processing Unit. That part of a computer system that controls the interpretation and execution of instructions. In general the CPU contains the following elements:

ALU: Timing and Control; Accumulator; *scratch-pad registers; program counter;* and address stack.

DEVELOPMENT SYSTEM — A system, usually designed around a microcomputer, that simplifies the design of hardware and software of a microcomputer system.

DISKETTE OPERATING SYSTEM — Low cost, high speed, high storage capacity peripheral system that substantially reduces the time required to assemble, edit and execute programs.

FLOPPY DISK — Flexible magnetic storage media (about the size of a 45-RPM record) that is used in *Diskette Operating System.*

HEX (HEXIDECIMAL) — Number system, using base 16, with numbers 0-9, A, B, C, D, E, F representing all 16 four-bit codes.

IN-CIRCUIT EMULATION (ICE) — Allows system designer to plug the powerful *Intellec MDS* resources into his system providing a rapid and convenient way to run, emulate and debug his system.

INTELLEC MDS — Microcomputer Development System designed to simplify the development of microcomputer systems based on the 8080 or series 3000 (bi-polar microcomputer family) or any future microcomputers that Intel develops.

INTERRUPT — An external event that temporarily suspends execution of the main *program* and causes a *subroutine* to be executed that "processes" the interrupting device; after completion of the interrupt subroutine, execution of the main program resumes.

I/O (INPUT/OUTPUT) — General term for the equipment that is used to communicate with the CPU and is not considered memory.

MEMORY — That part of a computer which holds instructions or data; normally the *program* is stored in ROM (read-only memory) or PROM (programmable read-only memory), while data is stored in RAM (random access memory).

MICROCOMPUTER — Computer whose *CPU* is a *microprocessor;* a microcomputer is an entire system with *CPU, MEMORY* AND *I/O.*

MICROPROCESSOR — CPU that is generaly comprised of one or several LSI elements.

OBJECT PROGRAM — Program which is the output of an *assembler* or highlevel language — usually considered to be the machine language of the particular computer.

PL/M (Program Language for Microcomputers) — A high-level language especially designed to optimize the generation of programs for a microcomputer such as the 8008 and 8080.

PROGRAM — Collection of instructions ordered in a particular sequence to perform some desired task; may be either a general purpose or, as in most microcomputer systems, a dedicated program.

PROGRAM COUNTER — A register in *CPU* which specifies the address of each instruction; normally incremented sequentially but can be altered with conditional branches or jumps.

SCRATCH-PAD REGISTERS — Registers contained on CPU that are used for storing intermediate or temporary results effectively reducing the number of program steps.

SIMULATOR — *Software* that simulates a particular computer allowing a programmer to debug his program prior to committing it to hardware.

SOFTWARE — Computer programs that are general-purpose programs such as *assemblers, compilers* and text editors.

SOURCE PROGRAM — Computer program written in a language designed for ease of expression by humans, e.g. *assembly language program.*

SUBROUTINE — Portion of user's program that may be called from several places within the main program or from an external *interrupt.*

WORD LENGTH — Number of bits in the computer word (usually 4, 8 or 16). ◻

Teleprinter Maintenance

(*Continued from page 20*)

converts these pulses into mechanical functions that control the codebar mechanism. Refer to figure 9.

The magnet coil is a 500 milli amp closed loop, between the selector magnet driver and the magnet. In the stop condition the output of the selector magnet driver is marking. This keeps the coil energized and the armature attracted to the magnet core. In the attracted position the armature blocks the start lever.

When a code combination is received the start pulse (Spacing) de-energizes the coil, and the armature drops away from the magnet.

No longer blocked, the spring biased start lever overtravels the armature, causing two thing to occur.

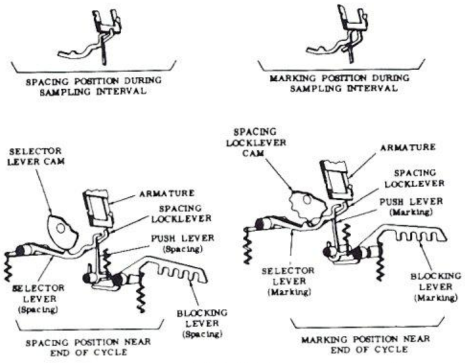
1. The start cam follower associated with the start lever falls into the indent of the start cam.

2. As the start cam follower falls into the indent, the trip lever associated with it moves away from the clutch shoe lever, allowing the selector clutch to engage.

The engaged selector clutch makes one complete revolution. The start cam follower remains in the indent of the start cam just enough to trip the clutch. It then comes out of the indent and rides the cam. This keeps the start lever away from the armature which will now be attracted if the incoming pulses are marking and unattracted if the pulses are spacing.

When the selector clutch rotates, the spacing locklever, the push lever reset bail, the codebar clutch trip follower arm, and eight selector levers ride individual cams under spring tension. Early in the codebar cycle the high part of the push lever reset cam pivots the push lever reset bail. In its motion the bail resets all the spring biased push levers selected in the previous cycle. Once reset the push levers can be positioned either marking or spacing as the intelligence pulses are received. Simply stated, we have a mechanical linkage between the armature and the trip lever on the selector clutch assembly. The selector clutch will respond only when an electrical impulse is felt in the selector magnet.

The action just described in the selector mechanism is really the guts of the Teletypewriter, the heart of the operation. We will find ourselves returning again and again to this area due to its complex functions. ◻



(Left Front View)

Selector Mechanism

Figure 9

Group Purchase For SCCS Members

A group purchase program has been established to enable all SCCS members to participate in the buying of selected products at discounted rates. To find out products and prices, a taped telephone message is prepared monthly, (213) 438-3890, with updated information. The tape states the date of the recording so callers can verify that they have current data. Announcements of available products will be made at each SCCS monthly meeting, and the tape is then updated. But please allow three days after the meeting for preparation of the new message before calling.

Since the message is stated only once and very quickly, it is suggested that a cassette recorder be used to make a permanent record of the information. The message will mention the product name and price only. For complete details on products listed, refer directly to the manufacturer or product vendor.

ORDER DEADLINE IS THE FIRST OF EVERY MONTH. Orders received on the first of the month prior to meeting date will be delivered at the meeting. Group purchase is based upon quantity requirement — if the minimum is not met, either a refund of your money, or holdover of delivery until the following month, may be required. Those who wish to receive shipment by UPS must add 5% to their purchase price to cover shipping and insurance. A 1% add-on (or $1, whichever is greater) is required to cover handling costs. California residents must add 6% sales tax to the total purchase price.

Send your check or money order, made out to the Southern California Computer Society Trust A, along with the order form, below.

GROUP PURCHASE ORDER FORM

Clip and mail this order form, with your check or money order, to

SOUTHERN CALIFORNIA COMPUTER SOCIETY
P. O. Box 987
South Pasadena, California 91030

Member Name ____________________

Membership # ____________________

Address ____________________
Street City State Zip

I will pick up my order at the next SCCS meeting ☐

Please ship my order to me, for which I have included 5% for shipping and insurance. ☐

QUANTITY	ITEM/DESCRIPTION	UNIT PRICE	TOTAL

Sub Total	
1% Handling ($1 minimum)	
6% Sales Tax	
5% Shipping and insurance	
TOTAL ENCLOSED	

(Continued from page 16)

Chapter I

PROGRAMMING

1. Introduction

A digital computer is capable of storing information, making calculations, making decisions according to the results of the calculations and arriving at the final solution of a given problem. However, it can not carry out these different tasks without being directed: each detail must be provided for it by the programmer. He therefore writes a programme, that is, a series of instructions which must lead the computer to a final solution; this series of instructions uses a method of calculation often known as algorithm.

2. The stages in writing a programme

In order to solve any problem with the help of a computer, the programmer must follow the five steps listed below:

a) definition of the problem to be solved;
b) determination of the best method of solution;
c) detailed analysis of the solution, flowchart;
d) coding of the solution in the programming language of the computer;
e) verification of the validity of the solution.

2.1. *Definition of the problem*

This is not always obvious but must be as complete as possible; thus, when a computer is to be used for process control, an extremely precise definition of the problem is necessary.

2.2. *Method of solution*

There are often several methods of solving a given problem; however, the best method must be chosen taking into account the possibilities of the computer employed.

2.3. *Selection and analysis of the solution*

The programmer has to select and analyse the solution, exhausting all possibilities that may arise, identifying the different steps necessary to solve the problem and arranging them in a logical order. A flowchart is a graphical means of representing the logical steps leading to a solution.

2.4. *Coding*

Coding, too frequently confused with programming, is only one phase of the procedure followed by the programmer. It consists in transcribing each instruction into its equivalent which can be understood by the computer. When the programme has been coded and stored in the memory, the problem can be solved; however, at this stage the programming is not usually finished; programmes which are written straight off and which work correctly at the first go are rare. In most cases, a fifth step is essential.

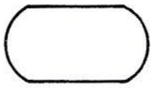
2.5. *Verification*

This requires the programmer to make a methodical step by step (instruction by instruction) analysis of the programme. To do this, the programmer usually has a verification or debugging programme available which allows the effect of each instruction in the programme to be seen.

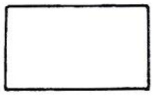
3. Flowchart

When a complex problem is to be solved, the programme requires numerous steps and writing them is tedious and time consuming. A method of solving a programme based on mathematical formulae is difficult to apply and the coding of the required instructions is not always evident. This is the reason for using a flowchart which represents the different steps needed. The essential purpose of the flowchart is to represent the logical sequence of operations and thus to greatly facilitate the coding of the programme. It is also a valuable help in a search for possible errors.

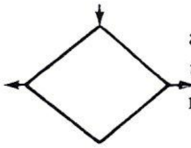
Graphically it is an arrangement of geometrical figures linked by lines with arrows indicating the direction of flow; the signification of the principal symbols is as follows:



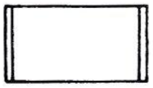
represents the start, the end or an interruption of the programme, depending on the word contained in the oval



represents a given task accomplished by the programme, the description of the task being briefly indicated inside the rectangle



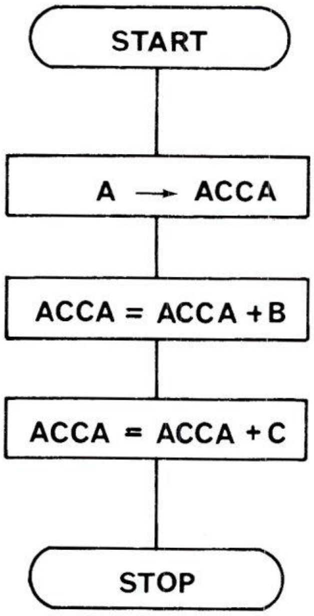
a diamond indicates that a test must be made to determine the subsequent path taken by the programme; the test is specified inside the diamond and its results marked above the appropriate output paths



represents one or several operations which are not detailed on the flowchart in question but are detailed on another flowchart; a sub-programme is often represented in this way.

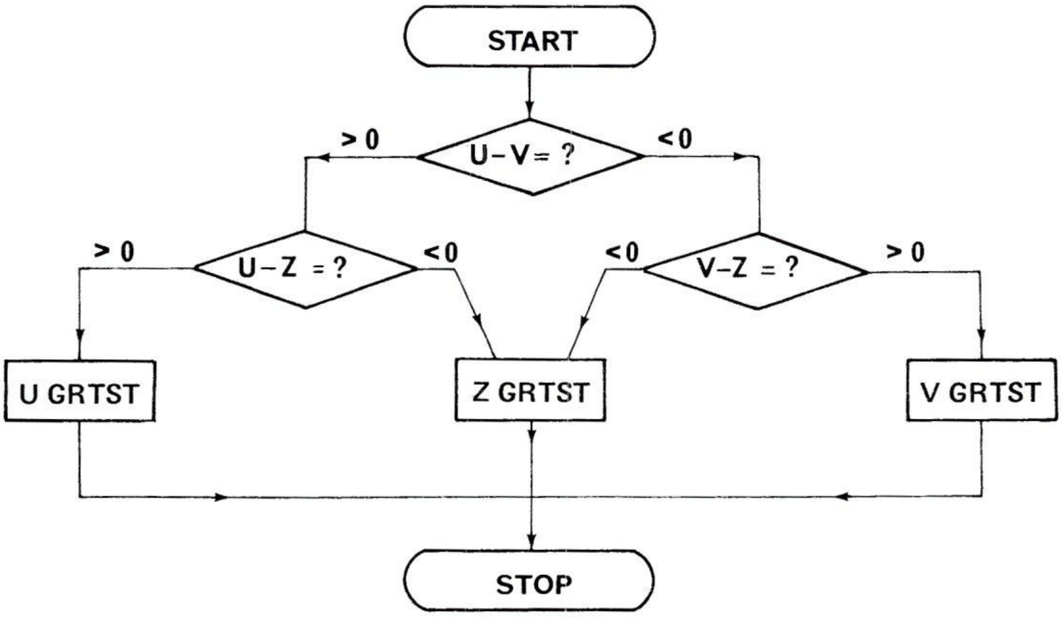
Example 1: Straight-line programming

The addition of 3 numbers A, B, C is represented by the following flowchart on which the direct sequential progression of the different steps may be noted.


START
A → ACCA
ACCA = ACCA + B
ACCA = ACCA + C
STOP

Example 2: branching

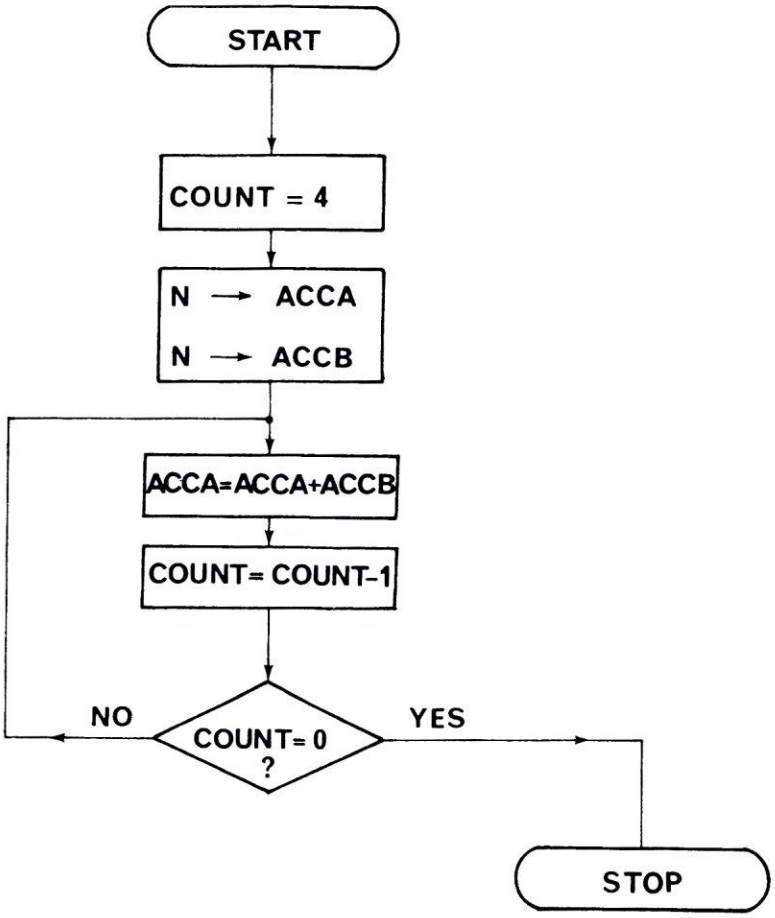
Search for the greatest of three rigorously different values U, V, Z.


START
> 0
U−V= ?
< 0
> 0
U−Z = ?
< 0
< 0
V−Z = ?
> 0
U GRTST
Z GRTST
V GRTST
STOP

Example 3: branch and loop

Given the binary value N, the value 5 x N is to be calculated. (It is assumed that the result may be represented in a single byte).

This example shows that there are at least two methods of solution; the first consists of moving N 2 places to the left (multiplication by 4) and then adding it to N; the second consists of adding N to itself 5 times; we will choose the second method.



4. Coding

The possibility of using an assembler programme which can be used with the MPU allows the programmer to write a programme in symbolic language using evocative mnemonic operation codes rather than translate each instruction into its binary, octal or hexadecimal form. Thus example 3 could be written in the following way provided that it is acceptable to store the first instruction at the address 0:

Address (octal)	*Instruction*	
0/	LDA A #4	load the counter to the value 4
2/	STA A 14	
4/	LDA A 15	load the two accumulators with (N)
6/	TAB	
7/	ABA	add the 2 accumulators
10/	DEC 14	decrease the counter by 1
12/	BNE 7	go to address 7 as long as the counter ≠ 0
13/	HALT	execution completed when the counter is at 0
14/	(COUNT)	
15/	(N)	

Writing the programme is greatly simplified by symbolic coding. However, writing the absolute address of each instruction is clearly not very practical as the number of bytes occupied by each instruction has to be known. In addition, if it is subsequently desired to add or destroy certain instructions it would be necessary to rewrite the instructions of which the operands make reference to the modified addresses. Moreover, if it is required to move the programme to another section of the memory, the programme must be completely rewritten with the new addresses. As such modifications are common, particularly in large programmes, there is a need for a simpler means of assigning memory locations: the assembler provides this facility.

4.1. *Assignment of the address*

As in the preceding example, most instructions are stored in sequential positions. If the programmer assigns an absolute address to the first instruction, the assembler is capable of knowing the addresses of the subsequent instructions. Thus, if it is required to store the instruction LDA A #4 at the address 0, this instruction must be preceded by the pseudo-instruction ORG 0 ("pseudo" because it is not translated into machine language).

4.2. *Symbolic addresses*

When he writes the programme, the programmer does not know which addresses he will use to store the variables of the problem; consequently, he would initially have to leave blanks after each instruction referring to the memory (example: DEC 14) and then fill them in once he had an idea of where to put the variables.

In fact, this is not necessary, as he has the possibility of assigning symbols to reference addresses and the assembler will replace these symbols by their numerical addresses.

With this facility, the programme 5*N may be written:

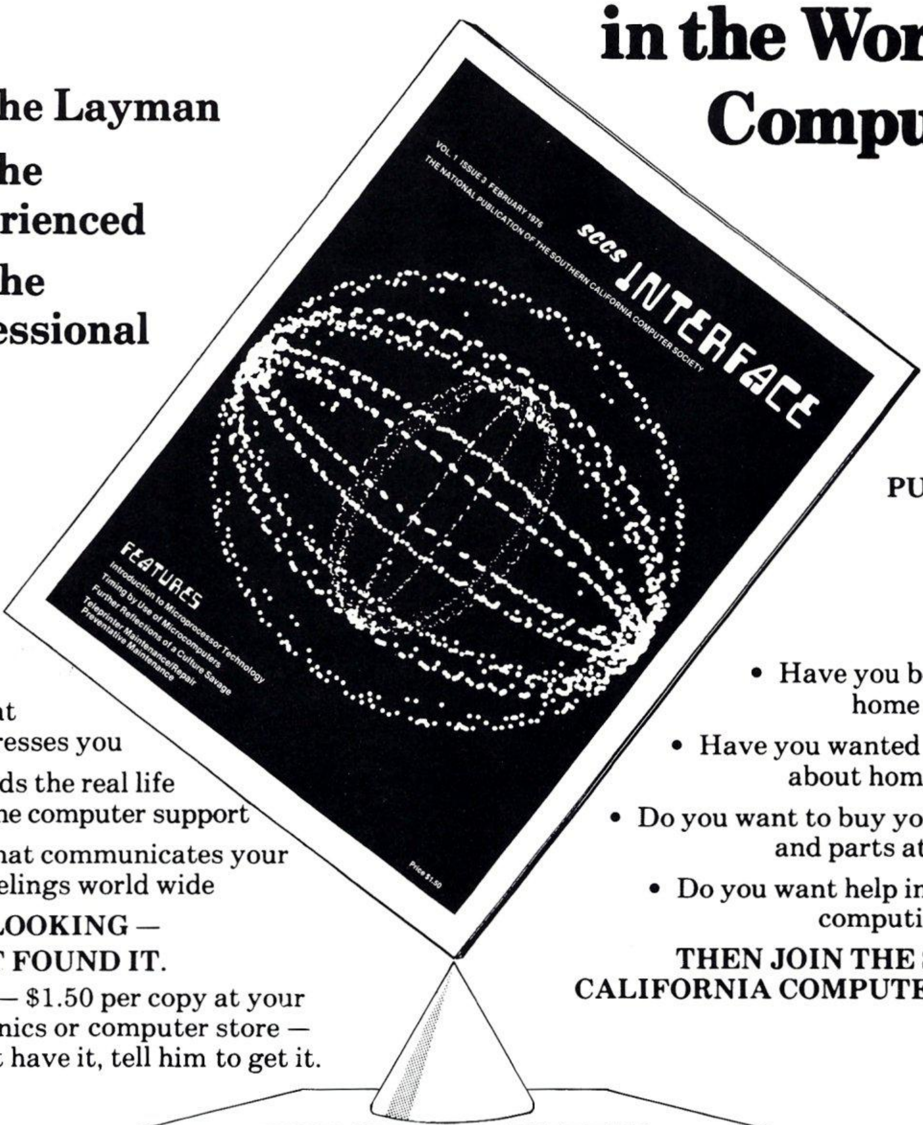
```
            LDA A #4
            STA A COUNT
            LDA A N
            TAB
NEXT        ABA
            DEC COUNT
            BNE NEXT
            HALT
```

TO BE CONTINUED NEXT MONTH

A Perfect Balance . . .

in the World of Computers

- **For the Layman**
- **For the Experienced**
- **For the Professional**


VOL 1 ISSUE 3 FEBRUARY 1976
THE NATIONAL PUBLICATION OF THE SOUTHERN CALIFORNIA COMPUTER SOCIETY
SCCS INTERFACE
FEATURES
Introduction to Microprocessor Technology
Timing by Use of Microcomputers
Further Reflections of a Culture Savage
Teleprinter Maintenance/Repair
Preventative Maintenance
Price $1.50

A MONTHLY PUBLICATION

- If you've been looking for a publication that personally addresses you
- That understands the real life problems of home computer support
- A publication that communicates your expression & feelings world wide

THEN STOP LOOKING — YOU'VE JUST FOUND IT.

INTERFACE — $1.50 per copy at your local electronics or computer store — if he doesn't have it, tell him to get it.

- Have you been alone as a home computerist?
- Have you wanted to learn more about home computing?
- Do you want to buy your equipment and parts at group rates?
- Do you want help in solving your computing problems?

THEN JOIN THE SOUTHERN CALIFORNIA COMPUTER SOCIETY

THE SUPER BENEFIT

JOIN THE SOUTHERN CALIFORNIA COMPUTER SOCIETY AND RECEIVE "INTERFACE" FREE.

DEAR READER:

Our purpose is to support the home computerist. To accomplish this we want to reach as far and wide as the postal service will allow. It makes no difference if you grow potatoes in Idaho, manufacture cars in Detroit, chase cows and oil wells in Texas, or simply enjoy retirement in Florida. You are invited to become a member of *SCCS*, a reader of *INTERFACE* and a member of that rapidly growing community of people who are discovering home computing.

ART CHILDS
Editor — *INTERFACE*

Book Review

(*Continued from page 34*)

mented. His simplification procedure works fine for all cases except where the low order byte is zero.

In the discussion of parity codes, no mention is made of the relatively common mark parity or of space parity.

A brief but helpful presentation regarding instruction codes will aid the reader in his understanding of later information.

In chapters 4, 5, and 6, the author presents a rather detailed discussion on a microprocessor CPU, its supporting logic, and basic programming procedures. Although given in somewhat generalized form, the information describes a hypothetical device with varying degrees of similarity to the microprocessors which are currently common. The reader may find this section somewhat difficult to absorb. Deciding just what statements are applicable to the microprocessor with which he is working may be beyond him.

I personally think it would have been better to go into detail regarding one of the more common microprocessors of the real world, with allusions to differences in other devices as need be.

Chapter 7 deals with the real world of microprocessor types, briefly considering the F-8, National's Pace and SC/MP, the 8080, the 6800, Rockwell's PPS-8, and the Signetics 2650. The degree of brevity is illustrated by a comparison of the over 200 pages dealing with the author's hypothetical computer with the 20 pages on the 8080 and the 12 pages on the 6800.

Although this chapter may give the reader a chance to compare some of the major characteristics of several of the more common microprocessors, it should not be as a major information source regarding any given device. Somewhat surprising is the omission of any discussion of multiple sourcing possibilities, a factor that I consider very carefully in all of my designs for clients.

It was interesting to read the comments regarding the future as seen through the author's crystal ball and his allusions to a 10-cent microprocessor. Having worked with transistors and other little solid-state marvels since 1948, and having seen the fantastic developments that seem to continue week after week, I refuse to even try to predict what the future holds (besides, I am having too much fun with the devices of the present!!).

An appendix lists some of the 7-bit ASCII (space parity) codes along with some of the EBCDIC character codes. With very little more effort or space, the table could have been made more meaningful and complete.

In summary, "Introduction to Microcomputers" is definitely a book worth adding to your library, whether you are a novice or a computer professional. It is definitely not a circuit "How-to" book, but contains a world of information in the limited space. It should prove profitable reading for both software and hardware types of computer nuts. ◻

THERE ARE TIMES WHEN YOU SHOULD IGNORE A HEART ATTACK VICTIM.

He says it's indigestion. Or he says it's nerves.

But if he has a steady pressure on the center of the chest lasting two minutes or more, ignore what he says. He could be having a heart attack.

One out of two heart attack victims denies that it's anything serious. Most wait almost two hours before seeking medical attention. By that time one-third of them are dead.

Don't you wait. If you think someone is having a heart attack, call the Fire Dept. Rescue Squad immediately. Seconds count.

Don't wait for severe pain, dizziness, fainting, sweating or shortness of breath. These symptoms may not occur. Unfortunately, a heart attack isn't always obvious before it's fatal.

Don't let fear of embarrassment delay your call.

If you're wrong about the existence of a heart attack, it doesn't matter. If you're right, nothing could matter more.

ACT FAST.
CALL THE FIRE DEPT. RESCUE SQUAD.
AMERICAN HEART ASSOCIATION
GREATER LOS ANGELES AFFILIATE

Unclassified Advertising.....

Caveat Emptor

Each month SCCS Interface will devote this space to free, non-commercial advertising by members. This is done as a service of the Southern California Computer Society and to help promote communications between computer experimenters and hobbyists throughout the world.

Free Ads for SCCS Members

To place your free ad, write or type your ad and mail to:

Nancy DeLong
8005 Denrock
Los Angeles, Ca. 90045

WHO NEEDS IBM? Are you using a DEC system-10? If your are interested in starting a club solely devoted to the 10, please call Ralph Klestadt at 784-8319 or 789-9616 after 4 p.m. Remember, you are not alone out there with your 10!

FOR SALE - MAYBE: Anyone interested in used hardcopy terminals (Syner Data Beta or Topograph) at $500-$600, call and leave phone no. or address. If we can get them we will let you know. ASCII Code, 10-15-30 cps, RS23C Interface, 132 column, adjustable with tractor, use back side of used printouts for your paper supply. Bill Kelly (213)379-2786.

FOR SALE-ALTAIR 8800 assembled and working. Consists of: 8¾K RAM, cassette I/O, Processor Technology 3P + S I/O, BASIC and assembler software. $1500. Robert Lasnik, 10328 Dunkirk Ave., Los Angeles, Calif. 90025.

MICROSYSTEM CONSULTANT for your special requirements for systems or hardware design. Ruben Loshak, 4040 Badillo Circle, Apt. 50, Baldwin Park, Calif.91706. (213) 338-8549.

WANTED TO BORROW or rent a Selectric for INTERFACE staff use. Everything you've read in this issue was prepared for type on a manual and my fingers are killing me! If you can help, please call Sheila, (213) 243-5179.

HP-65 USERS can get all white magnetic cards, 6 cards/dollar. Leonard Prince, 767 W. Rosslyn Ave., Fullerton, Ca. 92632 (714) 526-7881

NEED DEC Modules R401, W103, W501, W705, W706, W708 for PDP8/S and data for American Data Systems model 760 video terminal. Fred Schultz, 3734 W. Slauson Ave., Los Angeles 90043. (213) 299-4439.

WANTED: R-R tape recorder with slow speed, portable, for recording meetings. UHER or other. Also, pin feed TTY gummed labels. Louis Fields, 11662 Sunset Blvd. Suite 2, Los Angeles, Ca. 90049 [213] 472-0383

FOR SALE: KENNEDY Model 2869 9 TR, 800 BPI, Tape Drive. Has read/write capability at 15 IPS with 12 KB/SEC data transfer rate. Lists for $3,100.00 will sacrifice almost new for $2,100.00. Contact E. Wodrich at (213) 683-1560 Ext. 168. 9 AM-5 PM or (714) 622-9952 evenings.

WANTED For SCCS Headquarters. Furnishings for new office from desks and chairs to file cabinets and mics. tables, office machines. If you can donate or lend anything useful, please call Art Childs (213) 243-5179, or Larry Press (213) 399-2083, or Mike Dent (213) 677-5433.

ADVERTISING INDEX

Information Inquiry Number		Page
5	Arrowhead Computer-"The Computer Store"	17
8	BYTE	37
6	Computer Mart	25
3	Continental Specialties, Inc.	5
7	I.C. Electronic Supply, Inc.	35
10	IMSAI	C-III
2	MITS, Inc.	2
9	Polymorphic Systems	41
1	Southwest Technical Products, Inc.	C-II
11	Sphere Corp.	C-IV
4	Terminal Systems, Inc.	C-I, 9

NEW PRODUCT INDEX

91	A. P. Products, Inc.	32-33
92	EBKA Industries	32-33
93	Electronic Product Associates, Inc.	32-33
94	Industrial Electronic Engineers, Inc.	32-33
95	Oliver Audio Engineering	32-33
96	Sargent Distributing Co.	32-33
97	Systems Research, Inc.	32-33
98	Vector Electronics, Inc.	32-33

SCCS MEMBERSHIP NUMBER

Do you need to know your membership number? Perhaps you did know, but forgot it. You can find it — and be reminded each month — by looking on the SCCS INTERFACE mailing label.

INTERFACE BACK ISSUES

A limited quantity of December, January and February issues are available from the publisher. $1.75 per copy plus .50¢ for handling and postage.

Send Cash, Check or Moneyorder to:

McPheters, Wolfe & Jones
INTERFACE—BI
P.O. Box 1234
Cerritos, CA. 90701

Classified Advertising

MICROCOMPUTER RENTALS

Pro-Log-PROM Programmers
Intel-Intellec 8/MOD80
. . . And Others
Short-Term Rates Available

L.D. Bevan Company
1562 Devonshire Avenue
Westlake Village, CA 91361
(213) 889-3653 (415) 592-6933

ALTAIR
COMPUTER
REPAIR
AND ASSEMBLED SYSTEMS

pdc PACIFIC DATA COMPANY

5122 CANTERBURY DRIVE
CYPRESS, CA. 90630 714/821-1689

HOW TO DESIGN WITH MICROCOMPUTERS

(The most comprehensive three day microcomputer design course ever offered)

This course will provide you with the basics of designing with microprocessors. Numerous real examples and an objective examination of today's most popular microcomputer's features will be presented.

By Whom — Course conducted by Microcomputer Technique, Inc. and sponsored by Cramer Electronics.

Where — Los Angeles/Orange County area.

When — March 24, 25, 26

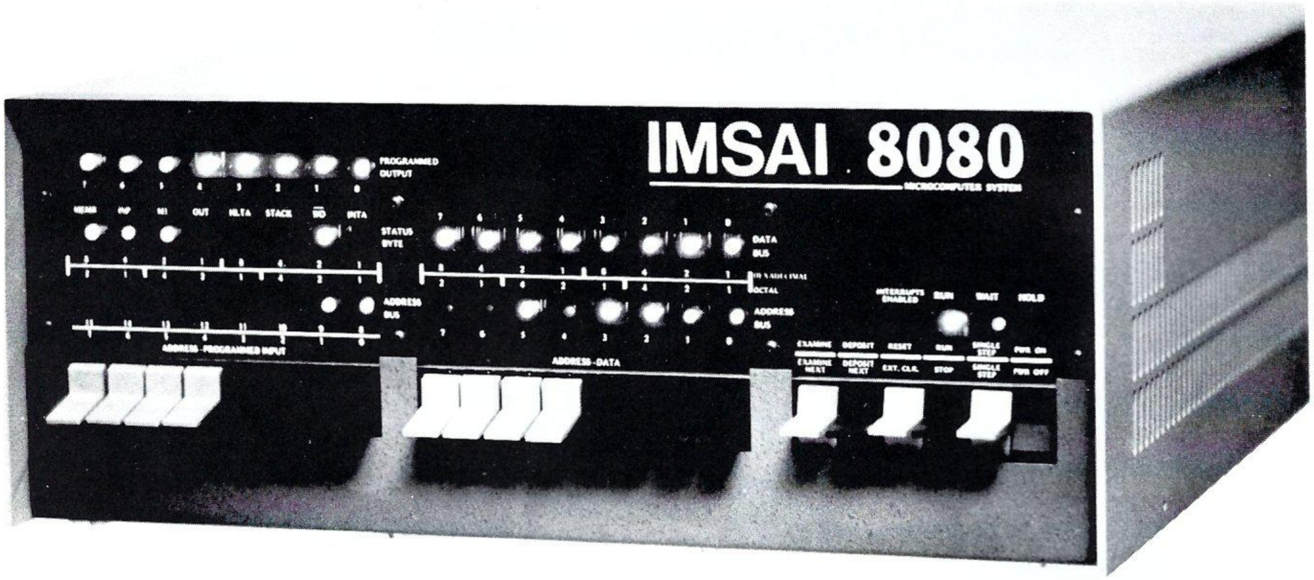
Cost — $395, includes manufacturers' technical literature and manuals, course notes, coffee breaks and luncheons for three days.

How to Register — Call Cramer/Los Angeles, 213/771-8300 or 714/979-3000, or contact your local Cramer salesman.

A. D. M. Communications, Inc.
TELETYPE & COMMUNICATIONS EQUIPMENT
BUY - SELL - RECONDITION
1322 INDUSTRIAL AVENUE
ESCONDIDO, CALIFORNIA 92025
(714) 747-0374
TELEX 695097

IMSAI®

COMMERCIAL GRADE COMPUTER – PRICED FOR THE INDIVIDUAL KIT BUILDER



QUALITY FEATURES INCLUDE:

A commercial grade cabinet, made of heavy gauge aluminum, custom designed for this specific product. Rack mounting is optional.

Lucite display panel with a photographically reproduced legend and LED display mask. In addition to standard lights and switches, the heavy duty front panel has an extra 8 program controlled LED's.

Rugged commercial grade paddle switches backed up by reliable debouncing circuits.

Gold plated contacts on board edge connectors to assure corrosion free contact at all locations.

A heavy duty power supply to handle lots of extra cards, the power supply delivers up to 20 amperes at 8 volts, and 3 amps each at +16 and -16 volts. A 30 ampere supply is optional.

A sturdy card cage to properly support up to 22 printed circuit boards.

Plug removable front panel board that eliminates the normal wiring harness between front panel and back plane. For special dedicated uses, the front panel board may be removed from the unit.

A full line of peripheral devices to build truly powerful and complete systems.

An exclusive shared memory facility from the company that pioneered shared memory microprocessing. Shared memory facilities allow multiple microprocessors in the same cabinet to share the same memory, yet run different programs in parallel with each other.

A full compliment of board options.

BASIC COMPUTER INCLUDES: 8080A Processor Board, Front Panel Control Board, lights and switches, power supply, Expander Board, Cabinet . . . and . . . Software (Assembler, Monitor, Text Editor, Loader and 4K BASIC).

PRICES: KIT . . . $499 ASSEMBLED . . . $931 4K RAM KIT . . . $165 ASSEMBLED . . . $299

OPTIONS: Rack mount, RAM & PROM boards, Parallel & Serial I/O Boards, an 8 level Priority Interrupt/Clock Board, Shared Memory Boards, Interface Boards.

PERIPHERALS: Floppy Disk Drives, 80 Column/300 LPM Printer, 50 Megabyte Disk Drive, 30 character per second Printer with Plotting Mode.

TERMS: Check or Money Order, Bankamericard, Master Charge, 25% deposit on COD orders. On all orders under $1,000, add 5% for handling. California residents add 6% tax.

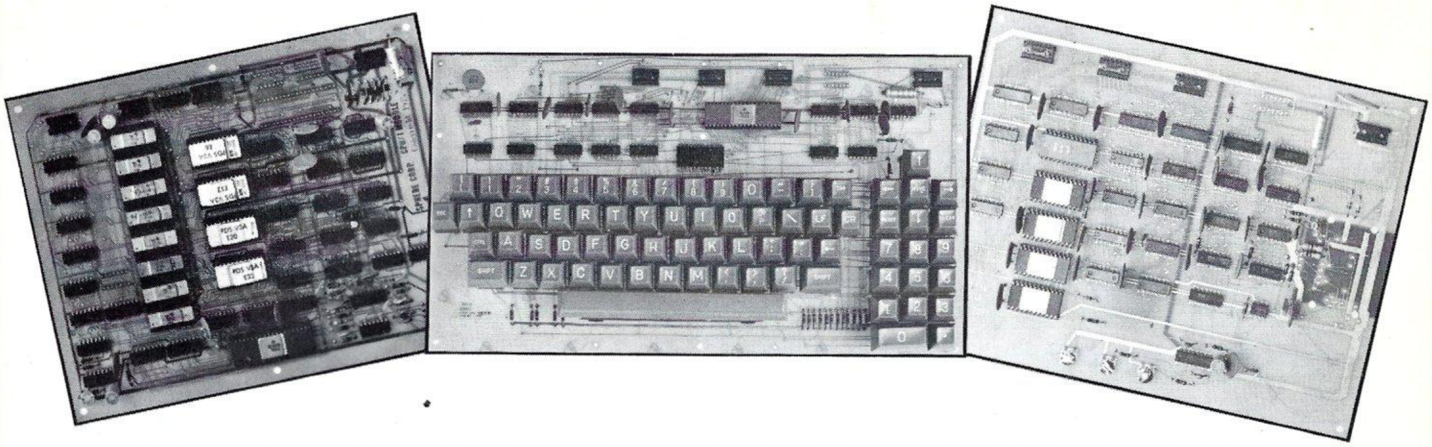
SEND FOR FREE CATALOG OF IMSAI MICROCOMPUTER PRODUCTS – DEALER INQUIRIES INVITED

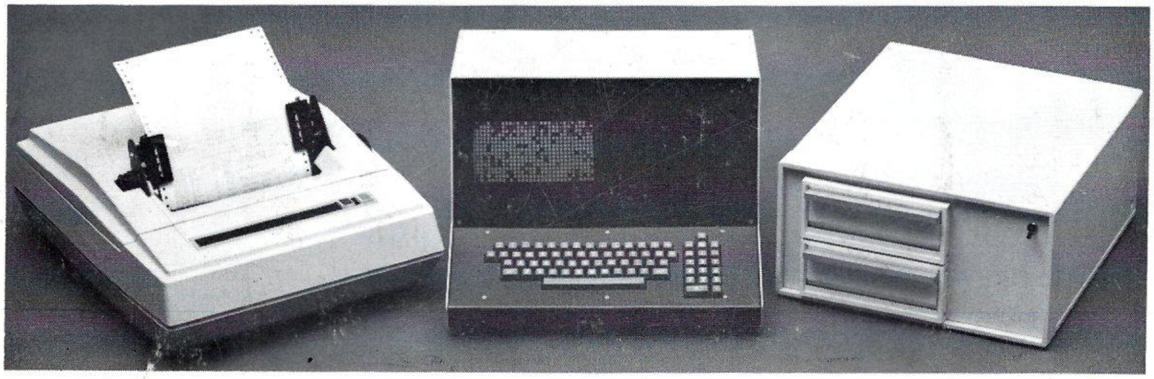
SPECIAL NOTICE TO ALTAIR 8800 OWNERS

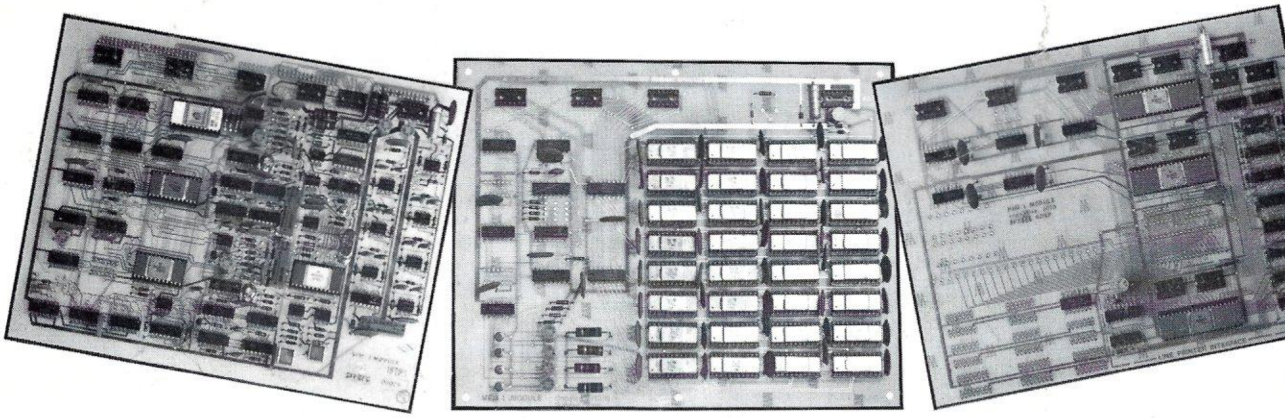
If you would like to step-up to the superior quality of an IMSAI 8080, you will be pleased to know that your ALTAIR 8800 boards are "plug-in" usable—without modification—in the IMSAI 8080 cabinet. Furthermore, by acquiring IMSAI's unique Memory Sharing Facility, your ALTAIR MPU Board and IMSAI MPU Board can co-exist in the same cabinet, operate in parallel with each other, and share all memory in common. This is the technology that laid the foundation for IMSAI's powerful HYPERCUBE Computer and Intelligent Disk systems (recently featured in Computerworld, Datamation and Electronics magazines.)

CIRCLE NO. 10 ON INQUIRY CARD

IMS ASSOCIATES, INC., 1922 REPUBLIC AVENUE, SAN LEANDRO, CA 94577
(415) 483-2093







...get it ALL together!

Please Contact One Of Our Distributors—

Computer Way—Huntington Beach, California
Bargain Electronics—LaMeda, California
Comput-O-Mat Systems—Rye, New York
The Computer Workshop, Inc.—Montgomery County, Maryland
Computer Mart Corporated—Boston, Massachusetts
The Computer Mart of New York—New York City, New York
Comunicacions S.A.—San Jose, Costa Rica
Computer Country—Denver, Colorado

SPHERE
CORPORATION

791 South 500 West Bountiful, Utah 84010
(801) 292-8466

DELIVERY: 60–90 DAYS